



POSGRADOS

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-19-No.277-2018

OPCIÓN DE
TITULACIÓN:

PROYECTOS DE DESARROLLO

TEMA:

DESARROLLO DE UN CONTROL DE
TEMPERATURA PI CON INTEGRADOR DE CLEGG
SINTONIZADO POR PSO

AUTOR:

JONATHAN RICARDO TIPÁN SIMBAÑA

DIRECTOR:

WILLIAM MANUEL MONTALVO LÓPEZ

QUITO - ECUADOR
2020

Autor:



Jonathan Ricardo Tipán Simbaña.

Ingeniero Electrónico mención en Sistemas Industriales.

Candidato a Magíster en Electrónica y Automatización, Mención en Informática Industrial por la Universidad Politécnica Salesiana – Sede Quito.

jtipans@est.ups.edu.ec

Dirigido por:



William Manuel Montalvo López.

Ingeniero Electrónico en Control y Automatización.

Magister en Control Electrónico y Automatización Industrial.

wmontalvo@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2020 Universidad Politécnica Salesiana.

QUITO – ECUADOR – SUDAMÉRICA

TIPÁN SIMBAÑA JONATHAN RICARDO

**DESARROLLO DE UN CONTROL DE TEMPERATURA PI CON
INTEGRADOR DE CLEGG SINTONIZADO POR PSO**

Contents

| | |
|---|-----------|
| Índice de Figuras | 5 |
| Índice de Tablas | 7 |
| Abstract | 8 |
| 1 Introducción | 10 |
| 1.1 Descripción general del problema | 11 |
| 1.2 Objetivos | 11 |
| 1.2.1 Objetivo general | 11 |
| 1.2.2 Objetivos específicos | 11 |
| 1.3 Contribuciones | 11 |
| 1.4 Organización del manuscrito | 11 |
| 2 Análisis y fundamentación teórica | 12 |
| 2.1 Estado del Arte | 13 |
| 2.2 Definiciones Previas | 16 |
| 2.3 Formulación del Problema | 28 |
| 3 Implementación y resultados | 29 |
| 3.1 Diseño del experimento | 30 |
| 3.2 Desarrollo del experimento | 30 |
| 3.2.1 Diseño de circuito de control usando Tarjeta STM32F407. | 30 |
| 3.2.2 Adquisición de datos del sensor de temperatura con LabVIEW y Matlab. | 32 |
| 3.2.3 Modelo matemático del sensor de temperatura en Matlab. | 33 |
| 3.2.4 Simulación de un controlador PI usando Matlab. | 36 |

| | | |
|-------|--|----|
| 3.2.5 | Sintonización usando aplicativo integrado PID Tuner de Matlab. | 38 |
| 3.2.6 | Diseño y Simulación del controlador PI+CI en simulink de Matlab. | 41 |
| 3.2.7 | Sintonización usando algoritmo PSO. | 44 |
| 3.2.8 | Implementación de controlador PI+CI en tarjeta STM32F407 | 47 |
| 3.3 | Resultados | 50 |
| 3.3.1 | Análisis Estadístico. | 55 |
| 3.4 | Discusión. | 56 |
| 3.5 | Conclusiones y recomendaciones. | 58 |

Índice de Figuras

| | | |
|------|---|----|
| 2.1 | Integrador de Clegg(CI) analógico [Prieur et al., 2018] | 16 |
| 2.2 | Respuesta a una entrada sinusoidal del Integrador de Clegg [Prieur et al., 2018] | 17 |
| 2.3 | Integrador de Clegg digital [Barreiro and Baños, 2012]. | 17 |
| 2.4 | Elemento reseteado de primer orden (FORE) [Barreiro and Baños, 2012]. | 18 |
| 2.5 | Estructura para un sistema basado en reseteo de primer orden (FORE) [Barreiro and Baños, 2012]. | 18 |
| 2.6 | Respuesta al escalón unitario de un sistema lineal reseteado [Barreiro and Baños, 2012]. | 19 |
| 2.7 | Diferentes respuestas sinusoidales del I+CI, para reset de cruce por cero ($e(t) = 0$) y para reset sectorial ($e(t)u(t) \leq 0$) [Barreiro and Baños, 2012]. | 20 |
| 2.8 | Componentes del sistema de control. | 21 |
| 2.9 | Partes del sistema de control de lazo cerrado o realimentado. | 22 |
| 2.10 | Curva de reacción a lazo abierto para una entrada escalón [Daful, 2018]. | 24 |
| 2.11 | Diagrama de flujo del algoritmo PSO. | 27 |
| 3.1 | Conexiones del entrenador de planta de control EPC [National Instruments, 2017]. | 31 |
| 3.2 | Diseño de circuito de control con STM32F407. | 32 |
| 3.3 | Diseño de conexiones para controlador con STM32F407. | 33 |
| 3.4 | Control lazo abierto, programación en bloques Waijung. | 34 |
| 3.5 | Panel de visualización y bloques de programación LabVIEW. | 34 |
| 3.6 | Sistema de Identificación Incorporado en Matlab. | 35 |
| 3.7 | Datos entregados por LabVIEW (derecha) , Datos importados a Matlab (izquierda). | 36 |
| 3.8 | Curva de comportamiento del sensor de temperatura del EPC. | 37 |

| | | |
|------|--|----|
| 3.9 | Estimación de función de transferencia de primer orden sin retardo. | 37 |
| 3.10 | Diagrama de flujo para la programación de respuesta al paso del sistema en lazo cerrado. | 38 |
| 3.11 | Simulación de control PI en el sistema. | 39 |
| 3.12 | Sintonización usando la herramienta PID Tuner de Matlab. | 40 |
| 3.13 | Sintonización en PID Tuner de Matlab sacrificando tiempo de establecimiento. | 41 |
| 3.14 | Sintonización en PID Tuner de Matlab sacrificando robustez. | 42 |
| 3.15 | Implementación de control PI+CI en simulink de Matlab. | 44 |
| 3.16 | Respuesta al paso unitario del sistema con control PI+CI en simulink. | 44 |
| 3.17 | Diagrama de flujo para la programación de respuesta al paso del sistema en lazo cerrado implementando calculo de p_{reset} | 45 |
| 3.18 | Diagrama de flujo de la función de costo para evaluar valores que encuentra el algoritmo PSO. | 46 |
| 3.19 | Diagrama de flujo de algoritmo PSO implementado. | 47 |
| 3.20 | Valores obtenidos por el algoritmo PSO en Matlab. | 47 |
| 3.21 | Respuesta al paso del sistema EPC con los valores encontrados por el algoritmo PSO | 48 |
| 3.22 | Programación de control PI+CI en simulink. | 49 |
| 3.23 | Botón de conexión para programar de forma correcta la tarjeta STM32F407. | 49 |
| 3.24 | Control de temperatura PI+CI implementado, visualización en LabVIEW. | 50 |
| 3.25 | Respuesta del sistema implementado con control PI sintonizado con PIDTuner. | 51 |
| 3.26 | Respuesta del sistema implementado con control PI sintonizado con PSO. | 52 |
| 3.27 | Respuesta del sistema implementado con control PI+CI sintonizado con PIDTuner. | 52 |
| 3.28 | Respuesta del sistema implementado con control PI+CI sintonizado con PSO. | 53 |
| 3.29 | Respuesta del sistema implementado con control PI+CI sintonizado con PSO rango ampliado. | 54 |
| 3.30 | Respuesta del sistema implementado con control PI vs control PI+CI. | 54 |

Índice de Tablas

| | | |
|-----|--|----|
| 2.1 | Comportamiento del sistema lineal con PID. | 23 |
| 2.2 | Reglas de sintonía ZN basadas en la respuesta al paso de la planta. | 24 |
| 2.3 | Reglas de sintonía CC. | 25 |
| 2.4 | Reglas de sintonía CHR. | 25 |
| 3.1 | Resultados de implementación de diferentes controladores y sintonizaciones | 55 |
| 3.2 | Muestras de valores ITAE obtenidas para análisis de test de Wilcoxon en el proceso de control de temperatura usando PI+CI | 60 |
| 3.3 | Análisis de resultados usando test de Wilcoxon. | 61 |
| 3.4 | Resumen de implementación de diferentes controles de temperatura. | 62 |

Resumen

El control de temperatura Proporcional e Integral-PI con integrador de clegg es un sistema de control basado en reseteo, el cual consta de un controlador lineal PI (Proporcional-Integral) al que se le ha incorporado un mecanismo de reseteo o reinicio a cero, conocido como el integrador de clegg(CI). El reseteo del controlador se genera únicamente cuando se cumple una condición. La condición que activa el reinicio o reseteo en este caso es cuando la temperatura del sensor alcanza el valor deseado o valor de referencia. También se puede decir que el reseteo se activa cuando el error entre la salida y la entrada es cero. Los valores de sintonización del control lineal PI se obtienen haciendo uso de la optimización por enjambre de partículas PSO. El fundamento del PSO simplemente es hacer que una población de valores de sintonización converja hacia la mejor solución posible para el control lineal PI. El propósito de este trabajo es implementar de forma práctica un sistema de control de temperatura basado en reseteo y sintonizarlo usando el algoritmo PSO. Seguidamente se evalúa el desempeño del controlador implementado basado en reseteo, frente a un control PI que es la alternativa clásica más utilizada. Finalmente los resultados muestran las propiedades que poseen los sistemas basados en reseteo frente a la alternativa clásica en una planta física real.

Palabras clave— Control Reseteado; Integrador de Clegg; PSO; Wilcoxon .

Abstract

The PI temperature control with clegg integrator is a reset-based control system, which consists of a PI (Proportional-Integral) linear controller with a built-in reset or zero reset mechanism, known as the clegg integrator(CI). The reset of the controller is generated only when a condition is met. The condition that triggers the reset in this case is when the sensor temperature reaches the desired value or reference value. It can also be said that the reset is activated when the error between the output and the input is zero. The tuning values of the linear PI control are obtained by using the PSO particle swarm optimization. The basis of PSO is simply to make a population of tuning values converge towards the best possible solution for linear PI control. The purpose of this paper is to implement a practical reset-based temperature control system and to tune it using the PSO algorithm. Then we evaluate the performance of the implemented controller based on resetting, against a PI control that is the most used classic alternative. Finally, the results show the properties of reset-based systems versus the classical alternative in a real physical plant.

Keywords— Reset Control; Clegg Integrator; PSO; Wilcoxon .

Chapter 1

Introducción

Al implementar un sistema de control lineal como el PI (Proporcional Integral) para algún dispositivo sea este: control de velocidad, de temperatura, presión, caudal etc., se encuentra que dicho sistema tiene limitaciones al controlar. Las limitaciones principales son: el ancho de banda y margen de fase, que se puede entender como el tiempo de establecimiento (rapidez) y la robustez respectivamente [Barreiro and Baños, 2012].

Estas limitaciones han llevado que muchos de los sistemas de control lineales utilizados, no satisfagan las necesidades de control requeridas. En este marco, muchos investigadores y grupos de trabajo han desarrollado varios métodos de solventar las limitaciones antes mencionadas. Uno de esos métodos es el control en base a sistemas reseteados, que con unas características de ganancia y ángulo únicos lograr solventar limitaciones que para los sistemas lineales sería muy difícil de superar.

El análisis y desarrollo de los sistemas basados en reseteo siguen en investigación, y cada año aparece una nueva forma de aplicarlo. Entre las principales formas de aplicación de los sistemas reseteados se tienen: la clásica, que condiciona el reseteo de cruce por cero del error, la ligada a sistemas impulsivos, donde el reseteo es de base temporal; y la enmarcada en sistemas híbridos que usan una condición sectorial [Barreiro and Baños, 2012].

1.1 Descripción general del problema

Analizando la información actual de los sistemas de control basados en reseteo, este tipo de controlador ha sido valorado generalmente mediante simulaciones y muy poco en procesos reales y mucho menos en procesos de control de temperatura, por lo cual es necesario analizar su comportamiento en una planta real y analizar su dinámica en un sistema de respuesta lenta como la que presenta el entrenador de planta de control - EPC en su sensor de temperatura.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollo de un control de temperatura PI + CI, sintonizado por PSO para evaluar el desempeño en un sistema real.

1.2.2 Objetivos específicos

- Caracterizar el lazo de control empleando manuales y diagramas para planificar la estrategia del controlador.
- Desarrollar el controlador PI con Integrador de Clegg empleando software especializado para analizar la respuesta transitoria del sistema.
- Desarrollar el algoritmo PSO del controlador, empleando software especializado para la sintonía óptima del controlador PI + CI.
- Evaluar el desempeño del controlador PI + CI con diferentes sintonías para determinar la fiabilidad del controlador.

1.3 Contribuciones

De este estudio se generan datos semilla que pueden servir como aporte a futuros trabajos de grupos de investigación del área del control automático, como el Grupo de Investigación de Electrónica y Automatización – GIECA de la Universidad Politécnica Salesiana.

1.4 Organización del manuscrito

Primero se presenta una base teórica sobre el panorama actual de los controladores basados en reseteo, el control lineal PI/PID hasta la optimización PSO. A continuación, se presenta el diseño e implementación de un control de temperatura haciendo uso de un controlador PI+CI, seguidamente se analizan los resultados obtenidos, para finalizar con las conclusiones y recomendaciones obtenidas al compararlo con un control clásico PI.

Chapter 2

Análisis y fundamentación teórica

En este capítulo se presenta lo más concreto posible el estado actual de varios y diferentes tipos de control de temperatura así como valores de comparación, de igual forma se indica los fundamentos teóricos necesarios para el entendimiento de los sistemas de control basados en reseteo. Finalmente se indica la problemática que se intenta resolver.

2.1 Estado del Arte

La idea del desarrollo de un control de temperatura lineal, incorporando un control reseteado como es el integrador clegg, nace en base a la investigación y estudio de las nuevas formas de control actuales, como una forma de encontrar un método que mejore las prestaciones de sistemas de control lineales y que sean fáciles de implementar. [Barreiro and Baños, 2012]

De la misma forma, los estudios de estos métodos de control, han ido acoplado diferentes herramientas en paralelo, como el algoritmo de optimización por enjambre de partículas - PSO, con el fin de encontrar soluciones a fenómenos estocásticos que surgen con el paso del tiempo en los diferentes sistemas a controlarse [Duarte and Quiroga, 2010].

El análisis y desarrollo de los sistemas basados en reseteo siguen en investigación, y cada año aparece una nueva forma de aplicarlo [Priour et al., 2018]. Entre las principales formas de aplicación de los sistemas reseteados tenemos: la clásica, que condiciona el reseteo de cruce por cero del error, la ligada a sistemas impulsivos, donde el reseteo es de base temporal; y la enmarcada en sistemas híbridos que usan una condición sectorial [Barreiro and Baños, 2012].

En la información recopilada hasta la fecha, no se tiene conocimiento del desarrollo de un control de temperatura con integrador de clegg, más sin embargo si existen varios métodos de control de temperatura utilizando diferentes formas de control, entre ellas se puede citar los siguientes:

- “Control de temperatura predictivo inteligente mediante el uso de PSO-RGA para procesos de calentamiento de moldes por transferencia en máquinas de envasado en semiconductores” [Tsai et al., 2017].
- “Control de temperatura basado en un algoritmo PID de neurona simple para una fuente de radiación de cuerpo negro”. [Wang et al., 2017a].
- “Diseño de controladores clásicos aplicados al control de temperatura para una máquina de termoformado de plástico” [Hernandez-Ambato et al., 2018].
- “Aplicación del algoritmo PID inteligente mejorado en el control termostático de alta precisión en el analizador de trazas de agua” [Qiao et al., 2018].
- “Control de temperatura del fluido mediante intercambiador de calor [Dulău et al., 2018].

De la misma forma se encuentra información acerca de implementación de controles basados en sistemas reseteados que se puede citar:

- “Diseño e implementación de un controlador basado en reinicio PI + CI en Emulador FPAA [Joraked et al., 2018].
- “Desarrollo de un algoritmo de semi auto-ajuste para el controlador PI + CI basado en reseteo” [Doktian et al., 2019].

Realizando una recopilación de los diferentes tipos de control de temperatura y las diferentes comparaciones que han realizado los diferentes autores, se puede tener una mejor idea de lo que se busca mejorar al controlar la temperatura de un proceso.

“Control de temperatura predictivo inteligente mediante el uso de PSO-RGA para procesos de calentamiento de moldes por transferencia en máquinas de envasado en semiconductores”[Tsai et al., 2017].

En esta investigación se encuentra una comparación con tres tipos de control aplicados al mismo sensor de temperatura con un punto de referencia de 70°C y unas condiciones de diseño de $\pm 3^\circ\text{C}$ de error en estado estable o régimen permanente. En una última comparación observan que el control PI+PSO+RGA que también es llamado control predictivo PI, es el más robusto y estable con un error de $\pm 1.37^\circ\text{C}$ en estado estable.

“Control de temperatura basado en un algoritmo PID de neurona simple para una fuente de radiación de cuerpo negro”[Wang et al., 2017a].

En este estudio se hace una comparación entre un PID clásico y un PID de neurona simple, mostrando las ventajas de esta innovadora forma de control sobre algunos controles clásicos. Enfatiza que la ventaja principal del PID de neurona simple es el ajuste de una sola ganancia K, mientras que el PID clásico necesita ajustar las tres ganancias clásicas K_p , K_i y K_d , además se centran más en la disminución del sobre impulso y la robustez antes que en el error en régimen permanente o mejorar el tiempo de subida y establecimiento, aseguran tener una alta precisión en régimen permanente.

“Diseño de controladores clásicos aplicados al control de temperatura para una máquina de termoformado de plástico”[Hernandez-Ambato et al., 2018]. En el cual se hace una comparación entre controladores on/off, PI clásico y un control PI con anti windup. Los autores indican que se determinó que el control PI con anti windup es el más apropiado para la implementación, debido a su bajo error en régimen permanente o precisión al controlar la temperatura. También hacen hincapié en la estabilidad del sistema una vez aplicado el controlador.

“Aplicación del algoritmo PID inteligente mejorado en el control termostático de alta precisión en el analizador de trazas de agua”[Qiao et al., 2018]. En este estudio se realiza un control de temperatura fuzzy con PID y se muestra sus ventajas de precisión en régimen permanente frente a un controlador PID clásico. El documento explica como se utiliza el control fuzzy con la lógica si-entonces para ajustar los valores de las ganancias del PID y obtener una precisión muy alta. Los autores concluyen con una misiva a mejorar su propio diseño para optimizar el tiempo de establecimiento y la robustez.

“Análisis y modelado del control de temperatura para el molino de crudo de una planta de cemento: un estudio de caso industrial que involucra el control PID”[Salamea et al., 2018].

En este desarrollo se enfatiza el uso de controladores clásicos para la industria

tomando como ejemplo la implementación de un control PID clásico, para la Cementera Nacional UCEM CEM Planta Industrial Guapán. La cual modelizan como sistema de segundo orden y sintonizan por el método de Cohen coon. En su diseño no especifican que error debe tener en régimen permanente, pero si indican que el punto de referencia de 70°C debe ser lo más estable posible. En su estudio afirman haber conseguido un control de temperatura preciso y robusto.

“Control de temperatura del fluido mediante intercambiador de calor” [Duláu et al., 2018]. Este estudio hace uso de un control on/off con histéresis, con una precisión en régimen permanente de $\pm 4^\circ\text{C}$ para varios puntos de referencia. El control on/off es realizado en base a la programación de un microcontrolador PIC15f1817 comunicado a un ordenador por comunicación RS232. Los autores prometen mejorar el control en base al control PID con criterios experimentales de sintonización.

“Análisis y Evaluación de la Eficiencia de un Sistema de Control de Temperatura Láser.”[Oo et al., 2019].

Este estudio hace uso del control PID y lógica difusa con reglas si-entonces para el ajuste de las ganancias del PID finalmente lo compara con un control PID clásico y con un control difuso puro. En el diseño del controlador no se especifica la precisión que se desea y únicamente nos muestra la respuesta al paso unitario del sistema controlado. Los autores indican que el mejor control finalmente es el control difuso puro debido a que no presenta sobre impulso alguno, sin tomar en cuenta el tiempo de estabilización o el tiempo de subida, que en el control PID es mejor.

“Diseño e implementación de un controlador basado en reinicio PI + CI en Emulador FPAA” [Joraked et al., 2018]. Este estudio se basa en simular el control de la temperatura de salida de una pequeña cámara de un disco duro a través de la corriente de una válvula de entrada de líquido refrigerante. El estudio muestra los resultados del uso del control basado en reseteo. Finalmente lo prueba en la planta virtual del sistema y los resultados se comparan entre el simulador SIMULINK y el emulador FPAA. Los autores indican que los valores obtenidos en ambas simulaciones son muy similares, pero que hay ciertas diferencias a destacar, como el tiempo de subida o el sobre impulso, que a pesar de implementar el mismo algoritmo no muestran exactamente los mismos valores. Los autores explican que esto se debe a los algoritmos de cálculo de cada simulador, así como el posible redondeo entre cálculos.

“Desarrollo de un algoritmo de semi auto-ajuste para el controlador PI + CI basado en reseteo” [Doktian et al., 2019]. Este estudio no se basa en el control de temperatura, sin embargo, muestra un método novedoso para el uso de controladores basado en reseteo, para una planta propuesta. Los autores realizan el estudio al comparar la variable K_r del integrador de cleeg, y observar que el controlador PI+CI ofrece un rendimiento excelente, que los controladores PI

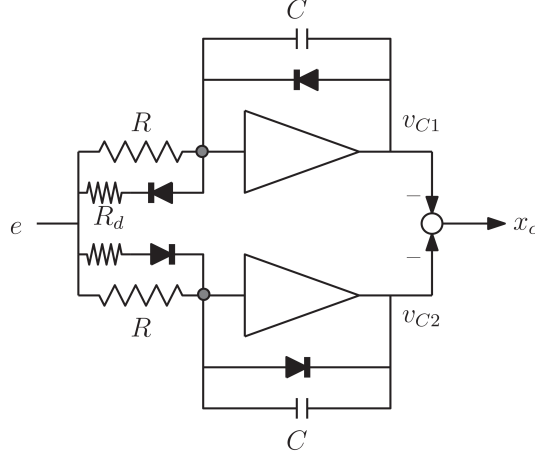


Figure 2.1: Integrador de Clegg(CI) analógico [Prieur et al., 2018]

ordinarios no pueden alcanzar.

2.2 Definiciones Previas

El controlador basado en reseteo.

La primera adaptación del concepto de control reseteado o basado en reseteo fue realizado por J.C.Clegg en 1958 es el conocido integrador de clegg CI o integrador reseteado [Clegg, 1958]. El integrador de clegg consiste en reiniciarse o resetearse cada vez que la señal de entrada cruza por cero, la Figura 2.1 muestra dicho integrador.

La dinámica del integrador Clegg es:

$$\begin{cases} \dot{X}c(t) &= \frac{1}{RC}e(t) & \text{si } e(t) \neq 0 \\ Xc(t^+) &= 0 & \text{si } e(t) = 0 \end{cases} \quad (2.1)$$

La primera ecuación en (2.1), representa la acción integral, la segunda ecuación representa la acción de reseteo en el siguiente instante luego de detectar el cruce por cero en la entrada [Barreiro and Baños, 2012].

En primer lugar, se debe tomar en cuenta que, por la suposición de ganancia infinita del amplificador operacional, los voltajes de entrada (marcados con puntos grises en la Figura 2.1) siempre son cero. Luego, los voltajes de los dos capacitores satisfacen $v_{C1}(t) \leq 0$ y $v_{C2}(t) \geq 0$ para todos los tiempos (de lo contrario, la corriente infinita que fluye en los diodos descargaría instantáneamente el capacitor). Además, cuando $e(t) < 0$, independientemente de la tensión anterior almacenada

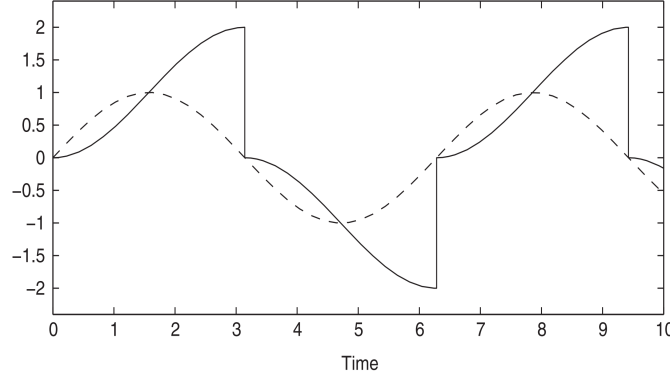


Figure 2.2: Respuesta a una entrada sinusoidal del Integrador de Clegg [Prieur et al., 2018]

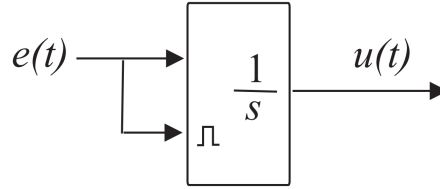


Figure 2.3: Integrador de Clegg digital [Barreiro and Baños, 2012].

en el condensador superior, la corriente que fluye en los dos diodos y a través del R_d superior impondrá (casi) instantáneamente $v_{C_1}(t) = 0$. Sin embargo, cuando $e(t) \geq 0$, el circuito superior corresponderá al integrador lineal porque ambos diodos estarán abiertos (se someterán a un voltaje no positivo). De manera similar para el circuito inferior, si $e(t) > 0$, se tendrá $v_{C_2}(t) = 0$ y si $u(t) \leq 0$ el circuito se integrará [Prieur et al., 2018]. En la Figura 2.2 puede observarse la respuesta del integrador de Clegg (línea continua) a una entrada sinusoidal (línea segmentada).

La idea del integrador de clegg fue abandonada hasta la década de los 70 retomada por Krishnan y Horowitz en 1974, quienes hicieron gran hincapié en las características frecuenciales (-20dB/dec con -38°) del integrador de clegg [Horowitz and Rosenbaum, 1975], el integrador de clegg puede conseguir prestaciones inalcanzables por controladores lineales [Prieur et al., 2018]. En el estudio tratado por Kridhnan y Horowitz se propone el elemento de primer orden llamado FORE, por sus siglas en inglés, pero no se especifica nada sobre estabilidad.

En la Figura 2.3 indica la representación del integrador de clegg obtenido en

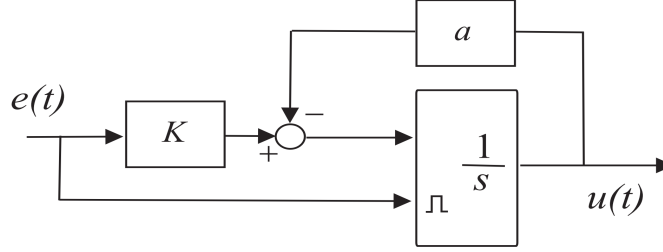


Figure 2.4: Elemento reseteado de primer orden (FORE) [Barreiro and Baños, 2012].

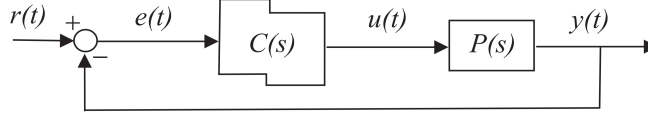


Figure 2.5: Estructura para un sistema basado en reseteo de primer orden (FORE) [Barreiro and Baños, 2012].

el entorno simulink habilitando una entra de reseteo externo [Barreiro and Baños, 2012], mientras que la Figura 2.4 representa el FORE.

En el segundo trabajo de Krishnan y Horowitz en 1975, se propone esta generalización (FORE), descrito por:

$$\begin{cases} \dot{u}(t) &= ke(t) - au(t) & \text{si } e(t) \neq 0 \\ u(t^+) &= 0 & \text{si } e(t) = 0 \end{cases} \quad (2.2)$$

En el modo continuo, el FORE tiene como función de transferencia $C(s) = \frac{k}{(s+a)}$, que se conoce como sistema de base lineal [Barreiro and Baños, 2012]. En la Figura 2.5 se muestra la propuesta de una estructura para un sistema de control basado en reseteo, con un bloque $P(s)$ y un bloque de rectángulo deformado que representa el bloque de reseteo del sistema de base lineal $C(s)$. Horowitz propuso una estrategia de diseño para el FORE en dos pasos [Beker et al., 2002].

- Partiendo de una planta $G(s)$, se diseña un controlador $R(s)$ que obtenga ancho de banda y rapidez de respuesta, aunque tenga excesiva sobre oscilación [Barreiro and Baños, 2012].
- Se inserta en lazo abierto $R(s)G(s)$ una pareja polo-cero, y se reagrupa de tal forma $[\frac{1}{(s+a)}][(s+a)R(s)G(s)]$, que se renombran $[C(s)][P(s)]$, y resetear

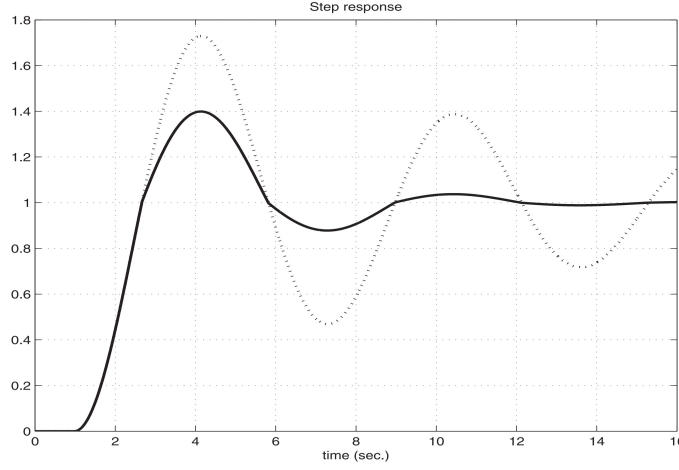


Figure 2.6: Respuesta al escalón unitario de un sistema lineal reseteado [Barreiro and Baños, 2012].

el primer factor, es decir convertirlo en un FORE [Barreiro and Baños, 2012].

En la Figura 2.6 muestra la respuesta de un sistema lineal incorporado con un control basado en reseteo, en el cual se puede observar que se preserva la rapidez del control lineal, pero mejorando mucho la robustez, al disminuir la sobre oscilación del sistema. Abandonada la idea nuevamente, no es sino hasta el 2001 en que se retoma y se realiza un avance considerable con el integrador de Clegg; el grupo de Y.Chait, planteó controladores de orden n y aportó diferentes resultados formales sobre estabilidad, como la condición H_β [Beker et al., 2004].

$$\begin{cases} \dot{x}_c(t) &= A_c x(t) + B_c e(t) & \text{si } e(t) \neq 0 \\ x_c(t^+) &= A_p x_c(t) & \text{si } e(t) = 0 \\ u(t) &= C_c x_c(t) + D_c e(t) \end{cases} \quad (2.3)$$

Como se mencionó anteriormente los estudios de Kridhnan y Horowitz no hacen hincapié en la estabilidad, por lo cual, los enfoques posteriores como el de Y.Chauti atacan principalmente ese problema de la estabilidad haciendo uso de representación en el espacio de estados y usando funciones de Lyapunov.

Las investigaciones posteriores se centran en sistemas de reinicio robustos basados en el modelo de (2.1) y la representación de bucle cerrado que surge de (2.3) [Priour et al., 2018]. La variedad de estudios realizados ha decantado en el desarrollo de diferentes tipos de control basados en reseteo. Como el reseteo de base temporal o impulsivos, los cuales abordan tres diferentes formar de enfoque que son:

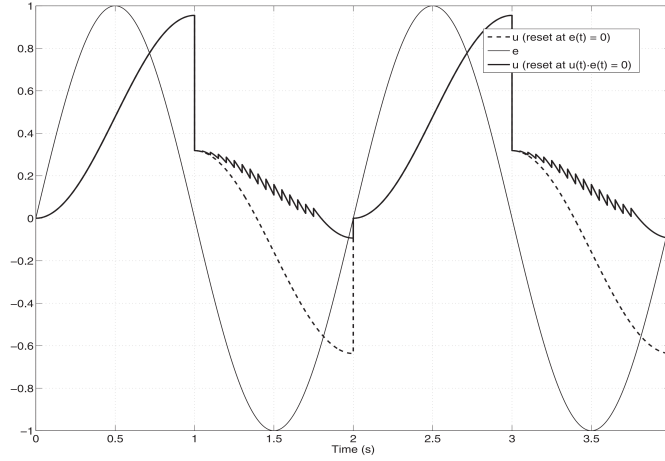


Figure 2.7: Diferentes respuestas sinusoidales del I+CI, para reset de cruce por cero ($e(t) = 0$) y para reset sectorial ($e(t)u(t) \leq 0$) [Barreiro and Baños, 2012].

- *Sistemas con impulsos tiempos fijos:* En el cual se tiene predefinido tiempos en los que se aplican las acciones de reseteo [Barreiro and Baños, 2012].
- *Sistemas con impulsos con tiempos variables:* En el cual el reseteo ocurre en subconjuntos arbitrarios del espacio producto tiempo \times estado [Barreiro and Baños, 2012].
- *Sistemas autónomos con impulsos:* Son aquellos en los que el reseteo se dispara por impacto del estado $x(t)$ sobre una cierta variedad M del espacio de estados, es decir, cuando $x \in M$ [Barreiro and Baños, 2012].

De igual manera se tiene control basado en reseteo de base sectorial que es basado en sistemas híbridos que, aunque mejoran la respuesta son propenso al castaño o “chattering” observable en la Figura 2.7. El reseteo sectorial ya no es equivalente al reseteo de cruce por cero. En general, no puede decirse que una modalidad de reseteo sea superior a la otra, simplemente son planteamientos diferentes que pueden tener ventajas particulares en cada caso [Barreiro and Baños, 2012].

De aquí en adelante, el interés por los CI y los sistemas basados en reseteo no hizo más que crecer, muchos autores y diferentes equipos de investigación contribuyen con aportes en el análisis, diseño y aplicaciones experimentales con los sistemas reseteados.

Control basado en reseteo y limitaciones fundamentales.

Desde las primeras propuestas, como el integrador de Clegg o el FORE, el

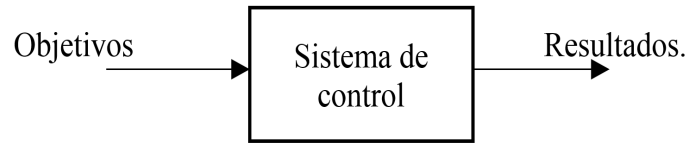


Figure 2.8: Componentes del sistema de control.

control reseteado tiene como motivación principal la búsqueda de mecanismos no lineales sencillos que permitan superar las limitaciones fundamentales lineales, la cuestión de la superación de limitaciones mediante reseteo todavía está lejos de ser clarificada, de una manera rigurosa y cuantitativa, y es una línea que merece investigaciones adicionales [Barreiro and Baños, 2012].

Las limitaciones fundamentales lineales, afectan, en el dominio temporal, a la velocidad de respuesta (tiempos de subida o de establecimiento) y a la sobre oscilación (o sub-oscilación, con ceros de fase no mínima)[Åström and Murray, 2010], de modo que, es imposible conseguir a la vez buena rapidez de respuesta y buen amortiguamiento, que resultan ser objetivos contrapuestos [Barreiro and Baños, 2012].

Control Clásico control PI y PID.

Durante las últimas siete décadas, los controladores PID se han utilizado con éxito en una variedad de aplicaciones industriales. En el control de proceso, la mayoría de los bucles son de tipos PI / PID [El-Shafei et al., 2017]. Los sistemas de control más utilizados en la industria y ampliamente aceptados por su simplicidad, son los controladores PI y PID (Proporcional Integral y Derivativo) [Salamea et al., 2018], debido a las grandes ventajas que presentan y sobre todo a la fiabilidad, su aplicación universal y la facilidad que tiene en el momento de su implementación [Tian et al., 2014], por esta razón estos controladores se han utilizado en los últimos años en varias aplicaciones industriales [El-Shafei et al., 2017].

El estudio de estrategias de control de temperatura es un tema importante en el control de procesos industriales[Wang et al., 2017b]. Al ser el control PI un sistema de control lineal, tiene componentes principales que destacar en la Figura 2.8 para el entendimiento general, estos son: objetivos de control, componentes del sistema de control, resultados o salida[Fernández, 2011].

Los sistemas de control se clasifican en dos tipos:

- Sistemas de control en lazo abierto: Aquellos en los que la variable de salida (variable controlada) no tiene efecto sobre la acción de control (variable de control) [Fernández, 2011].
- Sistemas de control en lazo cerrado (sistemas realimentados): Aquellos en los que la señal de salida del sistema (variable controlada) tiene efecto directo sobre la acción de control (variable de control)[Fernández, 2011].

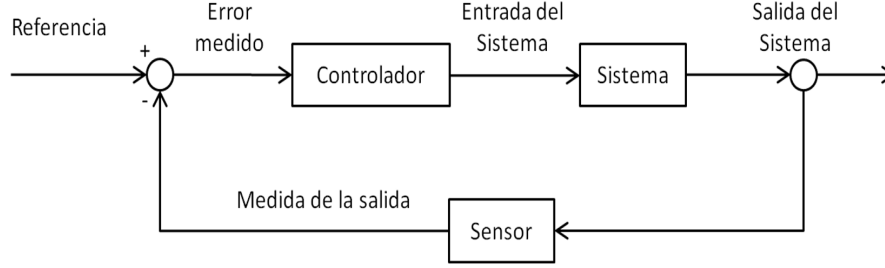


Figure 2.9: Partes del sistema de control de lazo cerrado o realimentado.

Un sistema con una o más trayectorias de realimentación como el de la Figura 2.9 se denomina sistema en lazo cerrado. Pero la realimentación no tiene efectos sólo en el error sino en características del desempeño del sistema como la estabilidad, ancho de banda, ganancia global y sensibilidad [Fernández, 2011].

El proceso para el desarrollo de un control PI o se un PID apenas y ha cambiado a lo largo del tiempo, por lo cual se puede decir que se ha estandarizado.

Como primer paso es el modelado matemático del equipo o planta que se desea controlar, esto se puede realizar de varias formas, las más actuales utilizan software matemático como Matlab, que parten de la obtención de datos medidos de la planta y los ingresan al software para obtener el modelo matemático más aproximado.

Otros métodos más tradicionales usan técnicas hechas a mano para modelar la planta como la aproximación de Taylor o Padé [Fernández, 2011]. Estas describen aproximaciones en base a modelos matemáticos por ejemplo para un sistema de primer orden con retardo como se expresa en (2.4):

$$G(s) = k \frac{e^{-\theta_0 s}}{T_s + 1} \quad (2.4)$$

Para los sistemas de segundo orden, la forma estándar del sistema de segundo orden viene dada por (2.5), [Basso, 2016]:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2.5)$$

Como segundo paso para el desarrollo del controlador es el uso de la ecuación que describe los tres parámetros del control PID también conocidas como ganancias como se expresa en (2.6).

$$Y(t) = k_p e(t) + k_i \int_0^t e(t) dt + k_d \frac{de(t)}{dt} \quad (2.6)$$

$Y(t)$ indica la salida en el dominio del tiempo del sistema, mientras que las ganancias k_p , k_i y k_d son las constantes de control: proporcional, integral y

Table 2.1: Comportamiento del sistema lineal con PID.

| Respuesta lazo cerrado | Tiempo de subida | Sobre pico | Tiempo de establecimiento | Error |
|------------------------------|---------------------|------------|---------------------------------|----------------|
| K_p | Disminuye | Aumenta | Poco cambio | Disminuye |
| K_i | Disminuye | Aumenta | Aumenta | Elimina |
| K_d | Poco cambio | Disminuye | Disminuye | Poco Cambio |

derivativo, respectivamente. Se aprecia que el error medido $e(t)$ está presente para cada ganancia, esto hace que de acuerdo al valor de las ganancias el error se corrige de manera distinta y la salida del sistema lineal manifieste un comportamiento diferente. Aplicando la transformada de Laplace para pasar (2.6) al dominio de la frecuencia como se expresa en (2.7) se obtiene:

$$Y(s) = k_p + \frac{k_i}{s} + k_d s \quad (2.7)$$

Seguidamente se realiza la elección del controlador de acuerdo a las necesidades; usualmente se utiliza más el control PI, pero obviamente existen casos en los que es preferible el uso del PID sobre todo en procesos que no poseen grandes retardos pero si presentan grandes desfases [Fernández, 2011].

Sintonización

La sintonización en controladores PI o PID, se refiere a encontrar valores constantes para las ganancias K_p , K_i y K_d , haciendo uso de diferentes métodos o técnicas muy empleadas [Ruíz, 2002], algunas de ellas son: la sintonización de Ziegler-Nichol, la sintonización de Cohen-coon, el método de ajuste CHR. Cada una de estos métodos tienen sintetizado su uso en tablas que facilitan el proceso de sintonizado. Cada ganancia afecta la respuesta en lazo cerrado del sistema de diferente manera, tal como se muestra en la Tabla 2.1.

Ziegler - Nichol: El método de ajuste de Ziegler - Nichol (ZN) determina el valor de k_p , τ_i y τ_d centrado en la respuesta transitoria de la planta [Daful, 2018]. Para obtener dichos valores, este método utiliza la curva de reacción obtenida de la respuesta de la planta en lazo abierto a una entrada escalón.

Las curvas de respuesta escalonada como la de la Figura 2.10 pueden generarse experimentalmente o a partir de una simulación dinámica de la planta [Daful, 2018]. La curva de reacción en forma de S se puede caracterizar por dos constantes, tiempo de retardo L y la constante de tiempo T , que se determina dibujando una línea tangente en el punto de inflexión de la curva y encontrando las intersecciones de la línea tangente con el eje de tiempo y la línea de nivel estable [Daful, 2018].

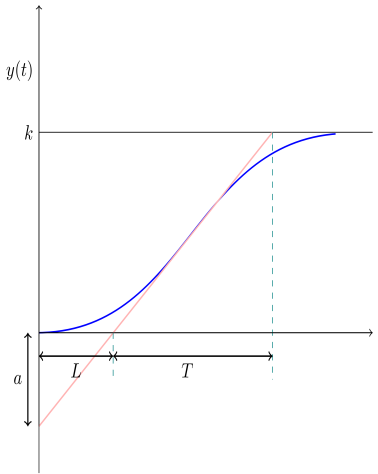


Figure 2.10: Curva de reacción a lazo abierto para una entrada escalón [Daful, 2018].

Table 2.2: Reglas de sintonía ZN basadas en la respuesta al paso de la planta.

| Modo de control | K_p | τ_i | τ_d |
|-----------------|-----------------|----------|---------------|
| P | $\frac{1}{a}$ | 0 | 0 |
| PI | $\frac{0.9}{a}$ | $3L$ | |
| PID | $\frac{1.2}{a}$ | $2L$ | $\frac{L}{2}$ |

Una vez obtenido los valores L , T y a se aplica las ecuaciones de la Tabla 2.2 para obtener los valores k_p , τ_i y τ_d .

Método de Sintonización Coen-coon (CC): En este método, se elimina la acción de control y se introduce un transitorio de bucle abierto mediante un cambio de paso de unidad en la señal al proceso. En la salida del elemento de medición, la respuesta al escalón se registra como curva de reacción del proceso como se mostró en la Figura 2.10[Daful, 2018]. El método CC es una versión más compleja del método ZN. En este método, la curva de reacción del proceso se obtiene primero, mediante una prueba de bucle abierto, ya que es más sensible que el método ZN. Los parámetros del controlador con este método están dados por

Table 2.3: Reglas de sintonía CC.

| Modo de control | K_p | τ_i | τ_d |
|-----------------|---|----------------------------------|--------------------------------------|
| P | $\frac{1}{a} \left[1 + \frac{0.35\tau}{1-\tau} \right]$ | 0 | 0 |
| PI | $\frac{0.9}{a} \left[1 + \frac{0.92\tau}{1-\tau} \right]$ | $\frac{3.3-3\tau}{1+1.2\tau} L$ | |
| PID | $\frac{1.35}{a} \left[1 + \frac{0.18\tau}{1-\tau} \right]$ | $\frac{2.5-2\tau}{1+0.39\tau} L$ | $\frac{0.37-0.37\tau}{1+0.81\tau} L$ |

Table 2.4: Reglas de sintonía CHR.

| Modo de control | con 0% de exceso | | | con 20% de exceso | | |
|-----------------|------------------|----------|----------|-------------------|----------|----------|
| | K_p | τ_i | τ_d | K_p | τ_i | τ_d |
| P | $\frac{0.3}{a}$ | 0 | 0 | $\frac{0.7}{a}$ | 0 | 0 |
| PI | $\frac{0.35}{a}$ | $1.2T$ | 0 | $\frac{0.6}{a}$ | T | 0 |
| PID | $\frac{0.6}{a}$ | T | $1.2T$ | $\frac{0.9}{a}$ | $1.4T$ | $0.47T$ |

fórmulas en la Tabla 2.3 donde $a = \frac{kL}{T}$ y $\tau = \frac{L}{(L+T)}$ [Daful, 2018].

Algoritmo de ajuste PID Chien-Hrones-Reswick (CHR): El método de ajuste CHR PID, también denominado método ZN modificado, se centra en la regulación del punto de ajuste y el rechazo de perturbaciones. El método proporciona una fórmula para el exceso de 0% y 20%. Las fórmulas de ajuste CHR PID se resumen en la Tabla 2.4 para la regulación del punto de ajuste. En comparación con el método ZN, la fórmula de ajuste de PID utiliza la constante de tiempo, T . La respuesta de circuito cerrado más amortiguada, que garantiza, para el modelo de planta ideal, la respuesta más rápida sin sobre impulso o con un rebasamiento del 0% , y la respuesta más rápida con un 20% de sobre impulso con un rebasamiento del 20% [Daful, 2018].

Sintonizado por optimización numérica: Este método se basa en la

disminución del error $e(t)$ producido en lazo cerrado, el cual es la diferencia entre el valor deseado de la variable controlada y el valor real [Orozco and Ruiz, 2003]. En otras palabras, la resta entre la entrada y la salida del sistema. De estos los más conocidos son los criterios integrales [Orozco and Ruiz, 2003].

- Integral de error absoluto.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (2.8)$$

- Integral del error absoluto por el tiempo.

$$ITAE = \int_0^{\infty} t |e(t)| dt \quad (2.9)$$

- Integral del error cuadrático.

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (2.10)$$

- Integral del error cuadrático por el tiempo.

$$ITSE = \int_0^{\infty} t e(t)^2 dt \quad (2.11)$$

Optimización

La optimización es un proceso de obtención de la mejor solución óptima para un problema [Selamat et al., 2013]. En este caso se usará este proceso para encontrar los mejores valores para la sintonización de los controladores lineales sean estos PI o PID [Montalvo López, 2015]. Se sabe que existen muchos métodos y procesos de optimización, pero en esta revisión únicamente se hace hincapié en el algoritmo de optimización por enjambre de partículas, por sus siglas en inglés PSO. James Kennedy y Russel Eberhart introdujeron por primera vez la optimización por enjambre de partículas en 1995. Es una técnica de optimización cuyo principio de funcionamiento se basa en el intercambio social de un enjambre, como en los bancos de peces o bandadas de aves [Selamat et al., 2013]. Su estructura se muestra en la Figura 2.11 y para esta revisión funciona fuera de línea, optimizando los valores de ganancias del controlador mediante la minimización de un parámetro de Error Cuadrado Integral (ISE) [Campos et al., 2018] o minimizando la integral de Error Absoluto por el tiempo ITAE por sus siglas en inglés [Joraked et al., 2018].

Los pasos del algoritmo de PSO se da como sigue:

- Inicializa un grupo de partículas, incluidas las posiciones aleatorias, las velocidades y la aceleración de las partículas.
- Evaluar la aptitud de cada partícula.

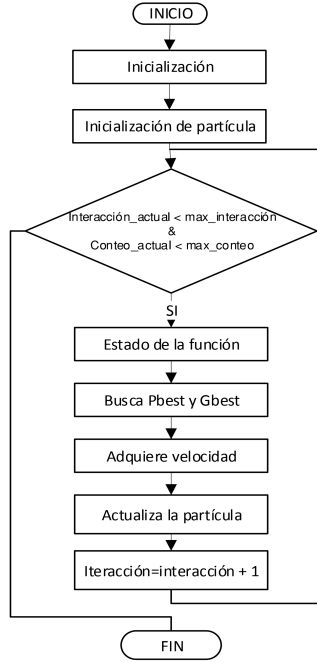


Figure 2.11: Diagrama de flujo del algoritmo PSO.

- Compara la aptitud individual de cada partícula con el Pbest anterior. Si es mejor, actualice como nuevo Pbest.
- Compara la aptitud individual de cada partícula con Gbest anterior. Si es mejor, actualiza como nuevo Gbest.
- Actualiza la velocidad y posición para cada partícula.
- Vuelve al paso 2 y repita todos los pasos hasta que se cumplan los criterios de fin.

El algoritmo PSO se describe matemáticamente por (2.12) y (2.13).

$$V^{k+1} = wV^k + c_1r_1(P_{best}^k + X^k) + c_2r_2(G_{best}^k + X^k) \quad (2.12)$$

$$X^{k+1} = X^k + V^{k+1} \quad (2.13)$$

Donde:

V^{k+1} : es la velocidad en la siguiente interacción.

w : factor de inercia determina el nivel de exploración de la partícula.

c_1, c_2 : factor de aceleración con valores entre 0 y 2.

r_1, r_2 : número aleatorio uniforme entre 0 y 1.
 X^k : distancia actual de la partícula.
 X^{k+1} : distancia en la siguiente interacción
 P_{best}^k : mejor posición individual de la partícula.
 G_{best}^k : mejor posición global de la partícula.

2.3 Formulación del Problema

El gran inconveniente actual que poseen los sistemas de control basados en reseteo, es que únicamente se los esta implementando de forma virtual, haciendo uso de diferentes software especializados. A pesar de que es un sistema que lleva muchos años en estudio, son muy pocos los sistemas de control basados en reseto que se han implementado físicamente. De igual manera la información de como se implementan en un equipo físico son muy reducidos, contrario a los métodos clásicos como el PI o PID, de los cuales se puede encontrar una gran cantidad de información, con muchas y variadas formas de implementación en el mundo real. De igual manera se puede decir de los métodos de sintonía de los sistemas de control, en específico del método de sintonía por algoritmo PSO, existe mucha información de las diferentes formas de hacer un algoritmo PSO pero la información de su aplicación directa en la sintonía de equipos de control es muy poca. Este artículo a demás de mostrar las propiedades de los sistemas reseteados, pretende ser una guía de implementación de un control basado en reseteo y hacer uso del algoritmo PSO para la sintonización del mismo. Así aportar a la comunidad con información que actualmente es limitada.

Chapter 3

Implementación y resultados

En este capítulo se indica en un primer apartado, el diseño del experimento en forma general, posterior se desarrolla paso a paso el diseño e implementación del control de temperatura PI+CI y su sintonía haciendo uso del algoritmo PSO. También se muestra los resultados obtenidos y se realiza una comparación con el sistema de control basado en reseteo frente al control clásico PI. Finalmente se indican las conclusiones y recomendaciones respectivas.

3.1 Diseño del experimento

El entrenador de planta de control EPC de DataLights, posee un sensor que mide la temperatura de un diodo emisor de luz LED, que al encenderse al máximo genera temperaturas que pueden llegar a los 100°C en 1 min aproximadamente. Al ser el LED el actuador o elemento electrónico que al encenderse genera en un área temperatura, esta magnitud es la opción a ser controlada, obteniendo los valores que suministre el sensor, para obtener un control robusto.

Para el desarrollo de un control de temperatura adecuado, se pretende hacer uso de un controlador PI agregando el factor de reseteo del Integrador de clegg, se plantea el uso de software especializado como lo es Matlab, con el cual facilite realizar los cálculos tanto de la parte de control, así como de la parte de sintonización y optimización de los valores adecuado para el controlador PI+CI, haciendo uso del algoritmo PSO.

Una vez logrado el controlador de forma virtual o en simulación, se pretende implementar de forma práctica, haciendo uso de una tarjeta electrónica programable con chip ARM, en este caso usando la tarjeta STM32F407. Esta última se programa haciendo uso el mismo software de desarrollo en este caso Matlab.

Para observar o visualizar como funciona el control y poder controlar las diferentes temperaturas que se deseen, se emplea el uso del software LabVIEW, a través de una tarjeta de adquisición de datos DAQ6009. Esto será una forma mas didáctica de observar el comportamiento del control PI+CI, y al mismo tiempo poder obtener valores para determinar el desempeño del controlador PI+CI implementado. Finalmente se compara el controlador PI+CI sintonizado por PSO, ante el software de control incorporado en Matlab, el PID Tuner, y así evaluar el desempeño del controlador PI+CI implementado.

3.2 Desarrollo del experimento

3.2.1 Diseño de circuito de control usando Tarjeta STM32F407.

Para diseñar cualquier circuito de control, lo primero es verificar como está constituido el sistema a controlar, por ende se verifica las conexiones del entrenador de planta de control (en adelante EPC). Se puede observar en la Figura 3.1 las diferentes conexiones que posee. En el caso del sensor de temperatura están muy bien señalados en la planta, tanto la entrada para el encendido del LED, así como para la lectura del sensor de temperatura.

Se verifica que el LED es un ON/OFF que funciona a 5v en lógica inversa, y el sensor de temperatura funciona entregando un voltaje entre 0 a 1 voltio [National Instruments, 2017]. Y usando la relación de temperatura y voltaje como se expresa en (3.1):

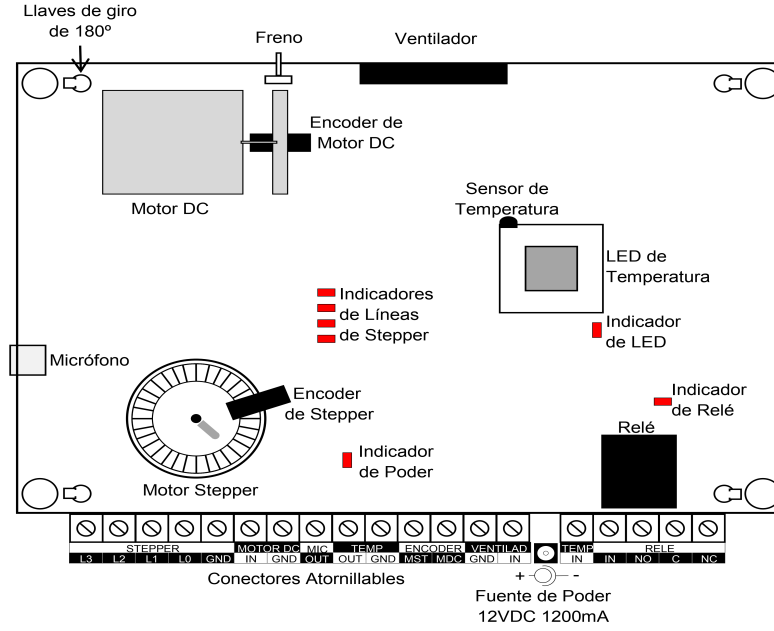


Figure 3.1: Conexiones del entrenador de planta de control EPC [National Instruments, 2017].

$$t = 100 \times v \quad (3.1)$$

Donde:

t : es la temperatura en grados centígrados.

v : es el voltaje en voltios

100: es el factor de relación de incremento

Para obtener la temperatura haciendo uso del voltaje se usa (3.1).

Con ello se puede diseñar un circuito con la tarjeta STM32F407, que haciendo uso en una salida digital, se obtenga $0v$ para encender el LED y $5v$ para apagarlo, y para leer los valores de temperatura se usa una entrada analógica. Se debe tener en cuenta que si se desea controlar la temperatura en función del encendido y apagado del LED, la mejor opción es usar la modulación por ancho de pulso ó por sus siglas en inglés PWM. El LED puede soportar frecuencias de hasta $100Hz$, con anchos de pulso observables de hasta $1ms$ [National Instruments, 2017]. Para poder observar los valores en un computador, se utiliza la tarjeta de adquisición de datos DAQ6009, con esta tarjeta se visualiza los datos de temperatura en pantalla y al mismo tiempo se puede enviar una señal de control para el punto de referencia o en inglés setpoint, tal como se observa en la Figura 3.2 y de una forma mas didáctica en la Figura 3.3.

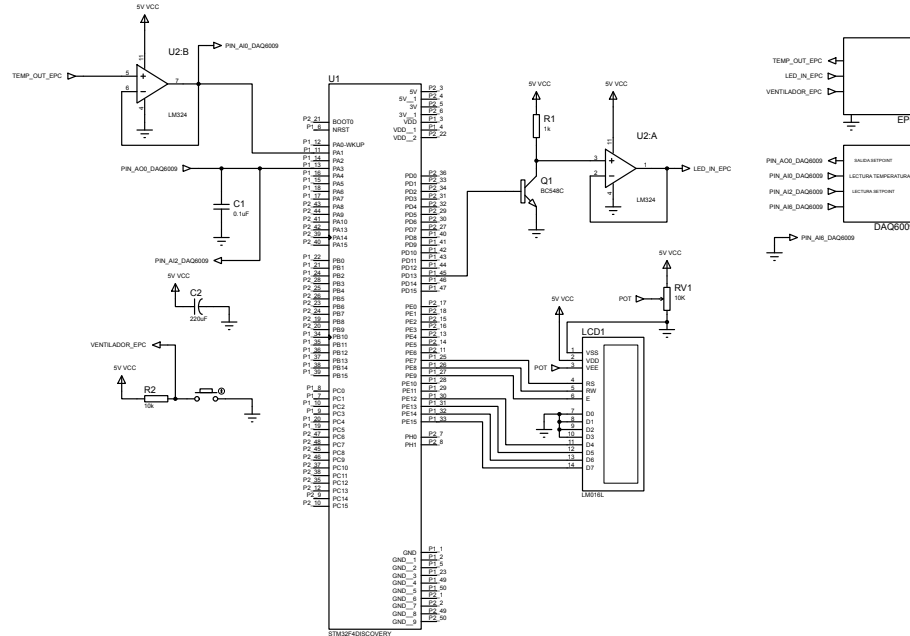


Figure 3.2: Diseño de circuito de control con STM32F407.

3.2.2 Adquisición de datos del sensor de temperatura con LabVIEW y Matlab.

Por medio de la tarjeta de adquisición de datos DAQ6009 se adquieren los datos del sensor de temperatura, y haciendo uso de LabVIEW se exportan a un archivo interno del computador, para poder trabajar con los datos adquiridos. Para llevar a cabo la adquisición de datos se debe hacer un programa para la tarjeta STM32F407, que genere pulsos PWM mas anchos o mas delgados, de acuerdo a un nivel de referencia.

Haciendo uso del conjunto de bloques de licencia gratuita Waijung de la compania tailandesa Aimagin Co.,Ltd, se puede programar fácilmente cualquier tarjeta de la familia STM32, desde simulink de Matlab [Aimagin Co., 2017].

Como se puede apreciar en la Figura 3.4 la programación basada en bloques, facilita y simplifica enormemente la programación en la tarjeta STM32F407. Con

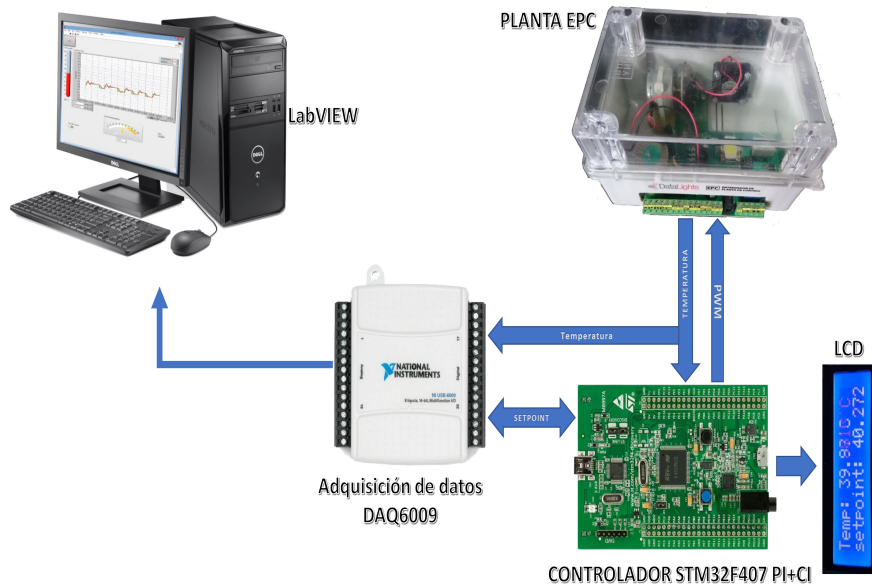


Figure 3.3: Diseño de conexiones para controlador con STM32F407.

este primer programa se puede controlar en lazo abierto la temperatura generada por el LED del EPC y adquirir los datos necesarios para el posterior modelamiento matemático.

En la Figura 3.5, se observa la programación realizada en LabVIEW para enviar la señal de control a la STM32f407, y también el valor de temperatura que se lee a través de la tarjeta de adquisición de datos DAQ6009.

Al mismo tiempo se hace uso de los bloques de LabVIEW, para almacenar los datos obtenidos en un archivo interno del computador, para este caso un archivo de hoja de calculo como se puede observar en la Figura 3.5. El cual debe ser almacenado en una ubicación ordenada y de fácil acceso.

3.2.3 Modelo matemático del sensor de temperatura en Matlab.

Existen muchas formas de modelar un sistema a controlar, siempre se busca la manera más rápida y práctica que se pueda utilizar, y así no perder demasiado tiempo en la parte matemática, que si bien es muy importante conocerla, se puede acelerar el proceso haciendo uso de las aplicaciones integradas en Matlab para este

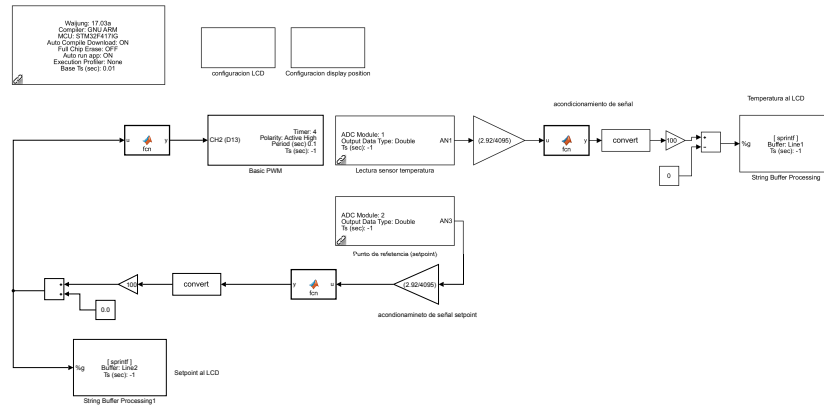


Figure 3.4: Control lazo abierto, programación en bloques Waijung.

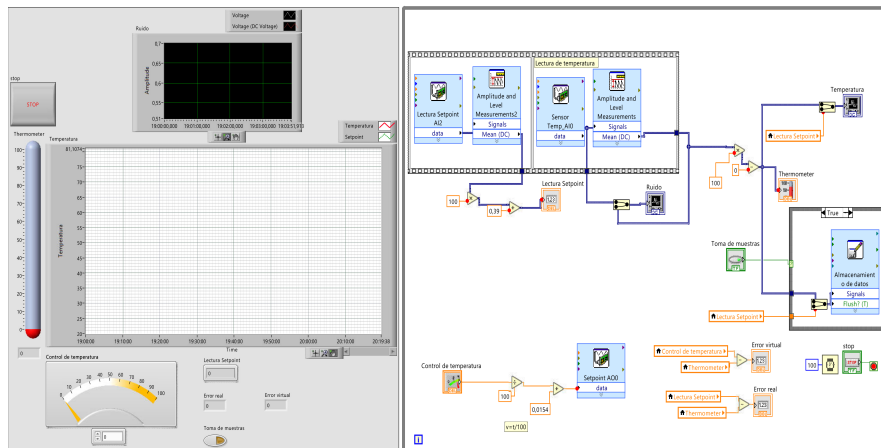


Figure 3.5: Panel de visualización y bloques de programación LabVIEW.

propósito. En este caso se hace uso del Sistema de Identificación de Matlab que se muestra en la Figura 3.6. Que haciendo uso de los datos obtenidos de un sistema

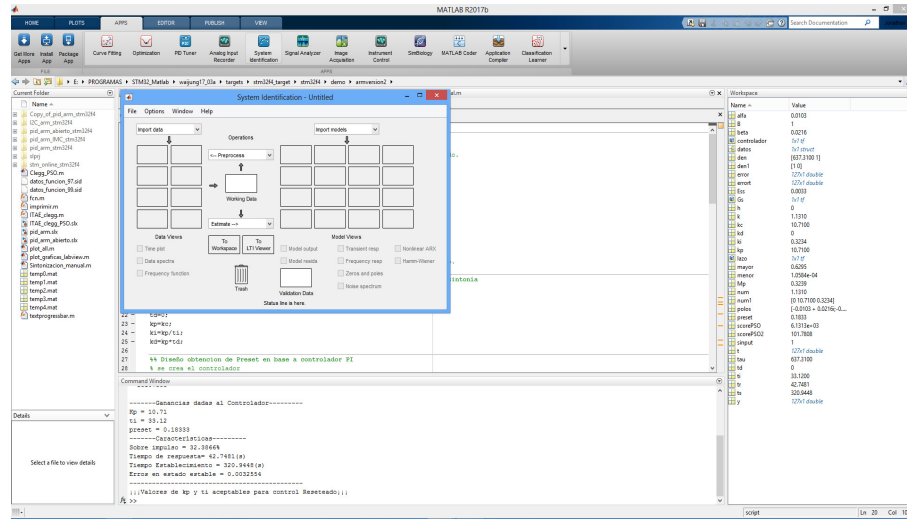


Figure 3.6: Sistema de Identificación Incorporado en Matlab.

que se desee controlar, se puede obtener el modelo matemático de dicho sistema con gran precisión. Así este modelo obtenido se puede utilizar en los diferentes entornos y aplicaciones integradas de Matlab.

Para realizar un modelado correcto, se tiene que analizar los datos que se estén utilizando. LabVIEW, almacena los datos con una marca de tiempo de horas, minutos y segundos como puede observarse en la Figura 3.7. Pero en Matlab, se pide tiempo de muestreo, por tal razón se debe transformar esa marca de tiempo obtenida en LabVIEW en tiempo de muestreo en segundos, usando (3.2).

$$tiempo\ de\ muestreo = \frac{(horas \times 3600) + (minutos \times 60) + segundos}{numero\ de\ muestras} \quad (3.2)$$

Este valor calculado debe ser ingresado al sistema de identificación de Matlab, seguidamente se observa en la Figura 3.8, la curva de comportamiento del sensor de temperatura en lazo abierto, generado en el sistema de identificación de Matlab. En el apartado “Estimate”, se selecciona “Process models”, se desmarca el retardo de tiempo “delay”, se selecciona “Disturbance Model”, “Order 1”, y se pulsa en el botón “Estimate”. Entonces calcula los parámetros para una función de transferencia de primer orden sin retardo de tiempo, como se observa en la Figura 3.9. Con esto se tiene la función de transferencia del sensor de temperatura del EPC en (3.3), la cual se usa para las diferentes simulaciones y cálculos necesarios para una correcta implementación.

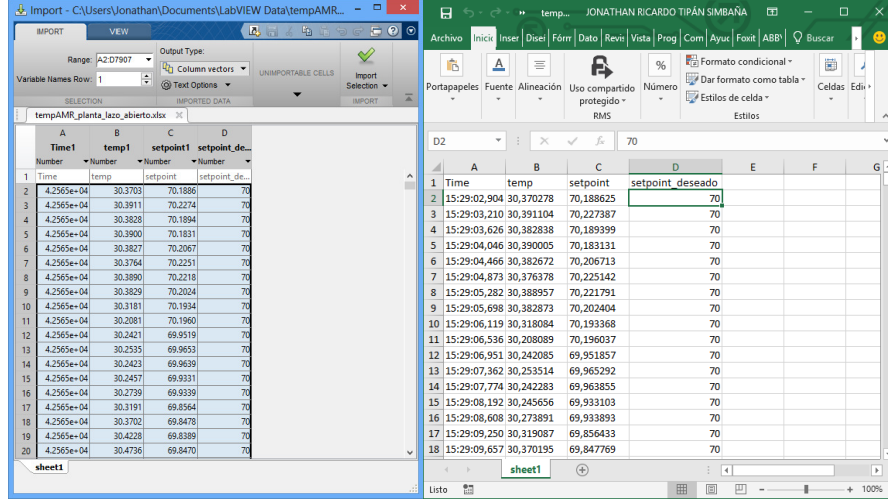


Figure 3.7: Datos entregados por LabVIEW (derecha) , Datos importados a Matlab (izquierda).

$$G(s) = \frac{1.131}{637.21s + 1} \quad (3.3)$$

$$G(s) = \frac{k}{\tau s + 1} \quad (3.4)$$

La función de transferencia obtenida, es de primer orden sin retardo, esto debido a que el EPC y su sensor se encuentran cerca y la lectura del sensor es inmediata o con retardo despreciable, pero Matlab puede generar funciones con retardo que es mucho mas común en el ámbito práctico profesional. Para obtener la función de transferencia con retardo, simplemente se debe marcar la opción “Delay”, en el sistema de identificación de Matlab. Al analizar la función de transferencia de 3.3, se puede decir que el sistema se estabiliza aproximadamente a los 637.21 segundos, debido a que τ indica el tiempo de establecimiento del sistema de primer orden.

3.2.4 Simulación de un controlador PI usando Matlab.

Con la función de transferencia del sistema a controlar, se necesita observar su comportamiento mediante simulaciones, en ocasiones es mejor realizar cualquier prueba de programación en simuladores antes de probarlos en la maquinaria implementada físicamente, y así poder evitar cualquier eventualidad no esperada. Para este fin se desarrolla un código en Matlab como en el diagrama de flujo de la Figura 3.10, para implementar un control PI en lazo cerrado a la función de transferencia que representa el sistema a controlar, tal como se indica en la Figura

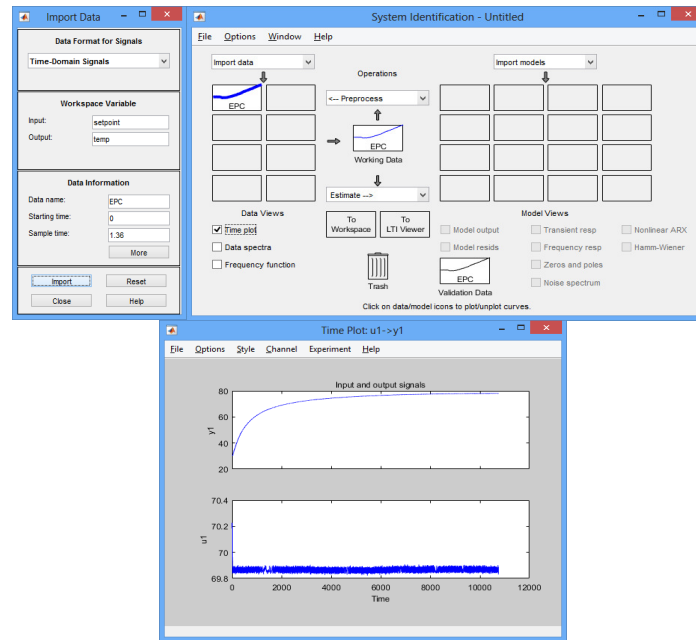


Figure 3.8: Curva de comportamiento del sensor de temperatura del EPC.

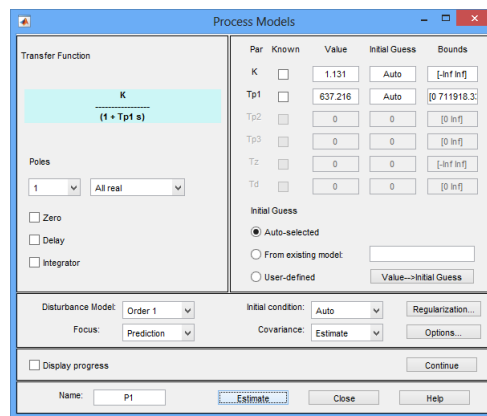


Figure 3.9: Estimación de función de transferencia de primer orden sin retardo.

2.9 del capítulo 2.

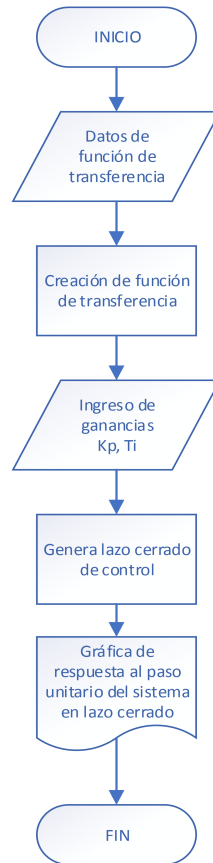


Figure 3.10: Diagrama de flujo para la programación de respuesta al paso del sistema en lazo cerrado.

El código genera la respuesta al paso de la función de transferencia, y lo muestra en una gráfica, cada vez que se modifique los parámetros o ganancias K_c , t_i , se estará sintonizando el controlador, y se observa en una gráfica el comportamiento del sistema. Como lo muestra la Figura 3.11 .

3.2.5 Sintonización usando aplicativo integrado PID Tuner de Matlab.

En el momento que se ejecuta el código de simulación del sistema de control de temperatura, la primera vez suele hacerse con valores aleatorios para las ganancias K_c y t_i .

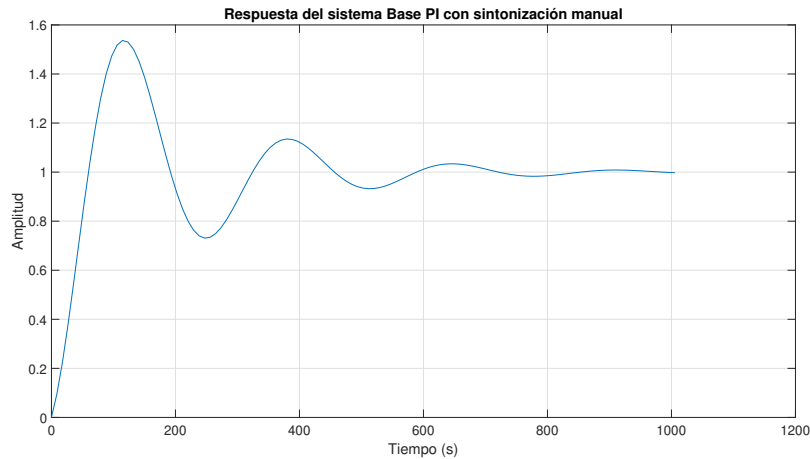


Figure 3.11: Simulación de control PI en el sistema.

En Matlab se encuentra una herramienta muy útil para la sintonización de sistemas de control clásico como son el PI y el PID, el “PID Tuner”, lo único que necesita esta herramienta es la función de transferencia del sistema que se desea controlar, y al ejecutar el código la función de transferencia queda en memoria o espacio de trabajo, por lo que se puede acceder desde el PID Tuner, una vez que la función de transferencia se encuentre en el espacio de trabajo de Matlab.

La finalidad de la herramienta PID Tuner, es encontrar valores de sintonización para las ganancias K_p , K_i , K_d o de forma estándar K_c T_i T_d , de los controles clásicos PI o PID. La herramienta PID Tuner posee barras de desplazamiento que facilitan la sintonización del sistema o planta a controlar, como se observa en la Figura 3.12.

En la Figura 3.12 se puede observar la función de transferencia del sistema siendo sintonizada de forma estándar, con las barras de desplazamiento ubicadas en la mayor velocidad y robustez que permite la herramienta PID Tuner. Al analizar detenidamente la Figura 3.12 la respuesta de sintonización, se observa, que con la mayor velocidad y robustez no alcanza el punto de referencia, en este caso el valor de 1, por ser la respuesta al paso unitario en el sistema.

Si se mueven las barras de desplazamiento, se puede obtener una mejor sintonización pero se tiene que sacrificar robustez o velocidad en tiempo de establecimiento como se observa en la Figura 3.13. Como se expresa en (3.3), el tiempo de establecimiento del sistema de control de temperatura en lazo abierto es de aproximadamente 637.21 segundos. Esto es mucho tiempo incluso en sistemas de respuesta lenta como los sistemas de control de temperatura.

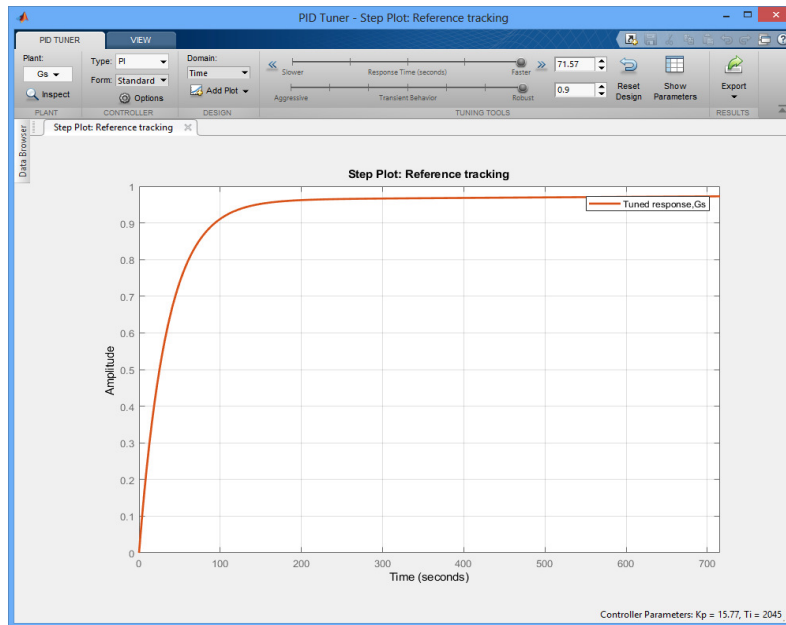


Figure 3.12: Sintonización usando la herramienta PID Tuner de Matlab.

Con el uso de sistemas de control se busca mejorar tanto la velocidad de establecimiento así como la robustez, es decir, que el sistema llegue rápidamente a la temperatura deseada y con la menor oscilación posible, para que el sistema de control pueda estabilizarse lo mas rápido posible. Pero estos objetivos suelen ser contrapuestos [Barreiro and Baños, 2012], como se observa en la Figura 3.13.

Se puede llegar al punto de referencia deseado con la máxima robustez posible pero con un tiempo de establecimiento de aproximadamente 2500 segundos, muy por encima del control en lazo abierto.

Por contra parte puede lograr un sistema mucho mas rápido, pero sacrificando robustez, por lo cual el sistema se vuelve oscilante como lo muestra la Figura 3.14, y si es muy oscilante acorta la vida útil de elementos eléctricos o electrónicos que se estén utilizando, o si es un sistema donde debe imperar la precisión en el control de temperatura, los fluidos o productos pueden echarse a perder.

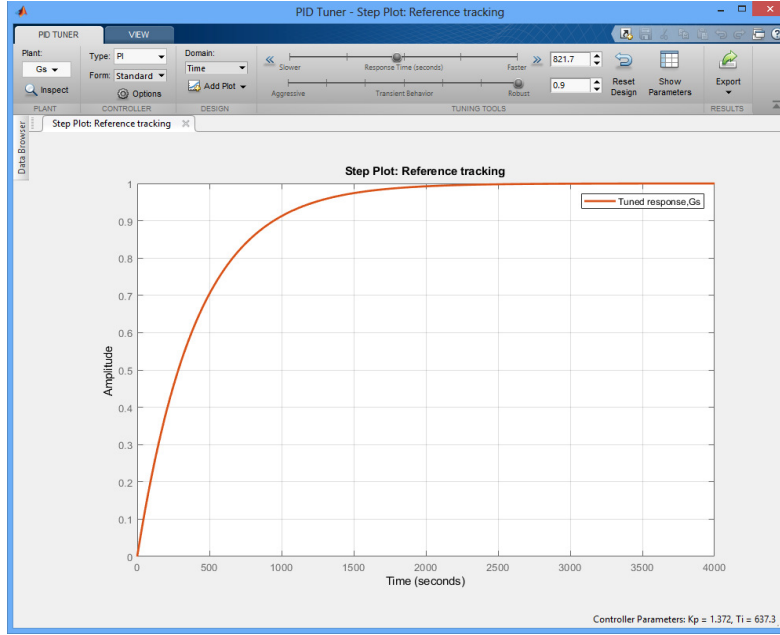


Figure 3.13: Sintonización en PID Tuner de Matlab sacrificando tiempo de establecimiento.

3.2.6 Diseño y Simulación del controlador PI+CI en simulink de Matlab.

Los controladores basados en sistemas reseteados logran romper la barrera entre el mejor tiempo de establecimiento y una gran robustez, que los controladores clásicos no pueden. Para el diseño del sistema de control reseteado PI+CI, se debe utilizar la ecuación que lo caracteriza.

En el dominio del tiempo se expresa en (3.5):

$$u(t) = k_p \left(e(t) + \frac{1 - p_{reset}}{T_i} \int_0^t e(\tau) d\tau + \frac{p_{reset}}{T_i} \int_{t_i}^t e(\tau) d\tau \right) \quad (3.5)$$

Donde P_{reset} es el porcentaje de reseteo que se da al controlador, tiene un valor entre 0 y 1, T_i es el tiempo integral, t_i es el instante de reseteo [Baños Torrico et al., 2010].

Si expresamos (3.5) en el dominio de la frecuencia se tiene (3.6):

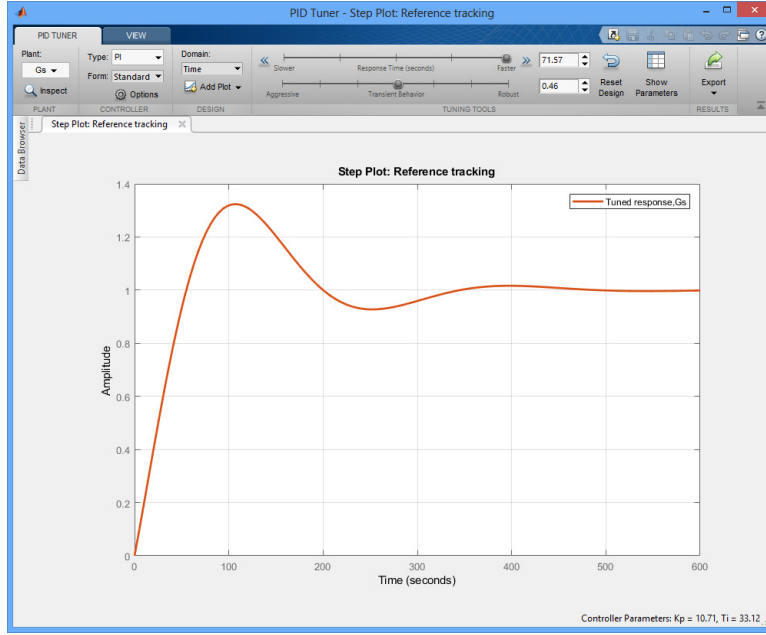


Figure 3.14: Sintonización en PID Tuner de Matlab sacrificando robustez.

$$(PI + CI)(s) = K_p \left(1 + \frac{1 - p_{reset}}{T_i s} + \frac{p_{reset}}{\cancel{T_i} s} \right) \quad (3.6)$$

donde $\frac{1}{\cancel{T_i} s}$, corresponde al integrador de Clegg que se expresa en (3.7):

$$CI(j\omega) = \frac{1.62}{\omega} e^{-j38.1^\circ} \quad (3.7)$$

Desarrollando el controlador PI+CI en el dominio de la frecuencia se tiene (3.8):

$$(PI + CI)(j\omega) = K_p \left[1 + \frac{1}{T_i} \left(\frac{1 - p_{reset}}{j\omega} + \frac{1.62 p_{reset}}{\omega} e^{-j38.1^\circ} \right) \right] \quad (3.8)$$

La Ecuación (3.8) [Doktian et al., 2019], se puede implementar código en diferentes lenguajes de programación, por facilidad, se utiliza simulink que ya tiene incluido el integrador que puede configurarse con reseteo externo como se observa en la Figura 3.15. Al simular usando la función de transferencia del sistema, agregando el control reseteado, y los valores correspondientes, se puede observar en la Figura 3.16; que el control PI+CI elimina la sobre oscilación y el tiempo de establecimiento es mucho mas corto de aproximadamente 200 segundos, mucho

menos que el control en lazo abierto y que el obtenido la herramienta PID Tuner.

Para sistemas de primer orden, el sobreimpulso puede ser eliminado después del segundo cruce por la referencia, es decir en el segundo instante de reseteo [Barreiro and Baños, 2012]. El grado de libertad adicional dado por p_{reset} del integrador de clegg, se usa para reducir el sobre impulso de la respuesta escalón unitario sin sacrificar velocidad de respuesta [Barreiro and Baños, 2012] y viene dado por (3.9):

$$p_{reset} = \frac{e^{-\frac{\alpha\pi}{\beta}}}{1 + e^{-\frac{\alpha\pi}{\beta}}} \quad (3.9)$$

siendo $\alpha \pm j\beta$ los polos de lazo cerrado del sistema.

Se debe tener en consideración que los valores de sintonización que se pueden obtener son infinitos, por lo tanto, para obtener una respuesta oscilatoria satisfactoria en un compensador, utilizamos un rango como se muestra en (3.10) [Barreiro and Baños, 2012].

$$\left(\frac{1 + kk_p}{2\tau} \right)^2 < \frac{kk_p}{\tau\tau_i} \quad (3.10)$$

Donde:

k : Ganancia en estado estable expresada en (3.4).

τ : Constante de tiempo dada por (3.4).

k_p : Ganancia proporcional del compensador o controlador obtenida usando algún método de sintonía.

τ_i : Ganancia integral del compensador obtenida usando algún método de sintonía.

Por lo tanto para el diseño de este controlador PI+CI; es necesario seguir ciertos pasos:

1. Ajustar el controlador base PI_{base} usando cualquier método de sintonía con valores k_p y T_i para una respuesta oscilatoria deseada usando (3.10).
2. Calcular α y β , que son la parte real e imaginaria positiva de los polos de lazo cerrado.
3. Finalmente se calcula p_{reset} usando (3.9).

Agregando el cálculo de los polos en lazo cerrado y de p_{reset} al código de Matlab, como se observa en la Figura 3.17, se puede utilizar los valores generados en espacio de trabajo de Matlab y llevarlos a simulink para poder simular el control PI+CI. En la Figura 3.16, se puede observar la respuesta al sistema con control PI+CI, usando los valores de sintonización obtenidos en la aplicación PID Tuner de Matlab. Al comparar visualmente las gráficas de resultados del control PI y del control PI+CI, se observa que disminuye el sobre impulso y las oscilaciones, sin sacrificar tiempo de establecimiento.

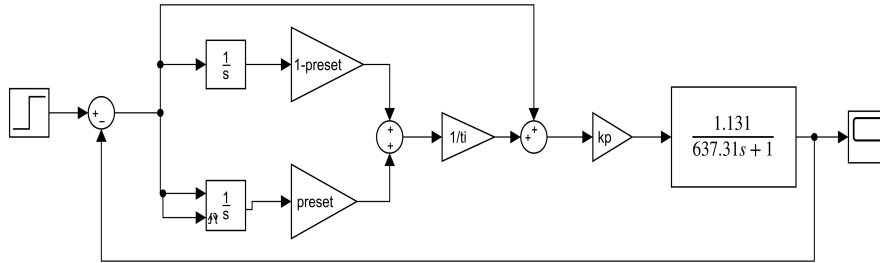


Figure 3.15: Implementación de control PI+CI en simulink de Matlab.

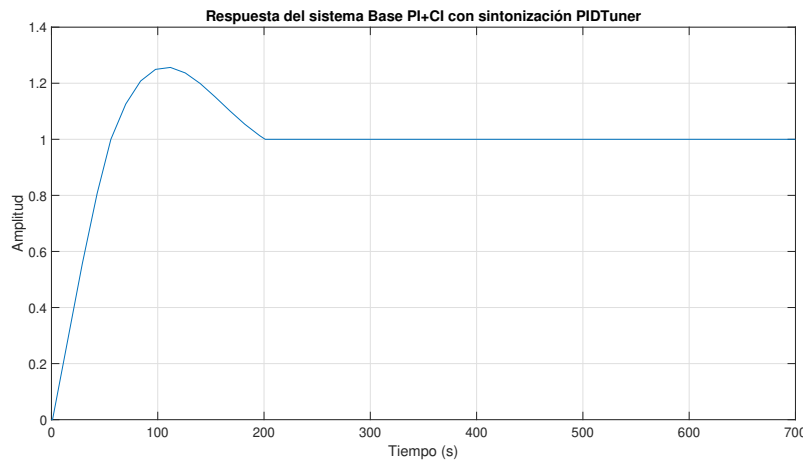


Figure 3.16: Respuesta al paso unitario del sistema con control PI+CI en simulink.

3.2.7 Sintonización usando algoritmo PSO.

Para mejorar la respuesta del sistema tanto en tiempo de establecimiento así como en robustez, es necesario encontrar valores de sintonización o ganancias apropiadas, pero puede convertirse en una tarea repetitiva que consuma gran periodo de tiempo si se los hace a mano. Por tal motivo se utiliza algoritmos que automáticamente realicen las tareas repetitivas de sintonización, encontrando el mejor valor posible, en un periodo de tiempo considerablemente corto.

El algoritmo de optimización por enjambre de partículas por sus siglas en inglés PSO, puede usarse para encontrar valores óptimos de sintonización. Al entregar

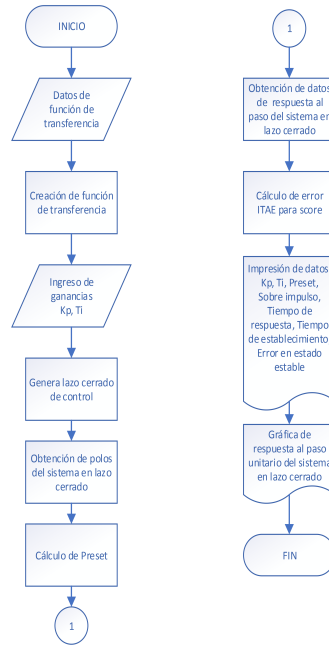


Figure 3.17: Diagrama de flujo para la programación de respuesta al paso del sistema en lazo cerrado implementando calculo de p_{reset} .

al algoritmo PSO valores de sintonización iniciales, busca y evalúa cada valor que encuentra en una función de evaluación o costo, que a su vez contiene la función de transferencia del sistema. La función de costo, devuelve un valor de puntaje o en inglés score, cada vez que se evalúan los valores de sintonización que encuentra el algoritmo PSO. En cada interacción del algoritmo PSO, el score va mejorando en la función de costo, hasta que finalmente cuando el algoritmo alcanza el máximo número de interacciones, se detiene.

Para obtener el score, se puede utilizar el método de sintonizado por optimización numérica, dado a conocer en el Capítulo 2, en este caso se usa la integral del error absoluto por el tiempo ITAE como se expresa en (2.9). Con esto se consigue evaluar el error en el transcurso del tiempo, es decir la diferencia entre el punto de referencia que se desea y lo que se obtiene realmente a la salida del sistema en cada instante de tiempo, por tal motivo, entre menor sea el score, menor será el error, el sistema tendrá menor consumo de energía y en consecuencia, mejor serán los valores de optimización.

La implementación del PSO sigue los mismos parámetros generales explicados en el Capítulo 2, cuyo algoritmo se representa en el diagrama de flujo de la Figura 2.11. Simplemente se realiza cambios en el uso que se le da al algoritmo, en esta

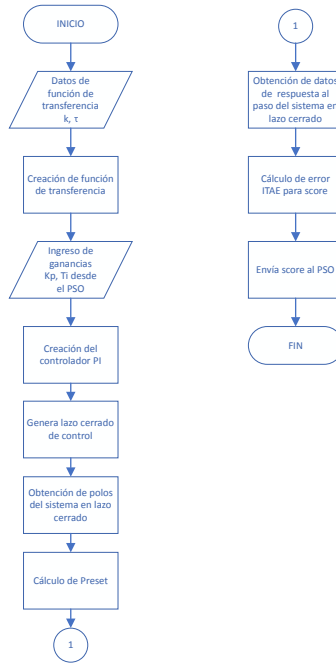


Figure 3.18: Diagrama de flujo de la función de costo para evaluar valores que encuentra el algoritmo PSO.

caso busca los mejores valores de optimización para las ganancias K_p y T_i , que necesita el controlador PI que a su vez, calcula el porcentaje de reseteo p_{reset} para el integrador de clegg.

En la Figura 3.18, se presenta el diagrama de flujo de la función de costo que utiliza el algoritmo PSO para evaluar los valores que encuentra en cada interacción. El diagrama de flujo del algoritmo PSO implementado se representa en la Figura 3.19.

La ejecución del algoritmo, genera valores dentro del rango de búsqueda o varianza en los valores iniciales de las ganancias K_p y T_i del controlador PI, y con esto calcula p_{reset} , para el integrador de clegg, al utilizar los valores encontrados y simular, se muestra la efectividad que tiene esta forma de sintonía. En la Figura 3.20, se observa los valores encontrados por el algoritmo PSO con una rango de búsqueda de $\pm 20\%$ para cada ganancia, y en la Figura 3.21, se tiene la respuesta del sistema con los valores encontrados por el algoritmo PSO.

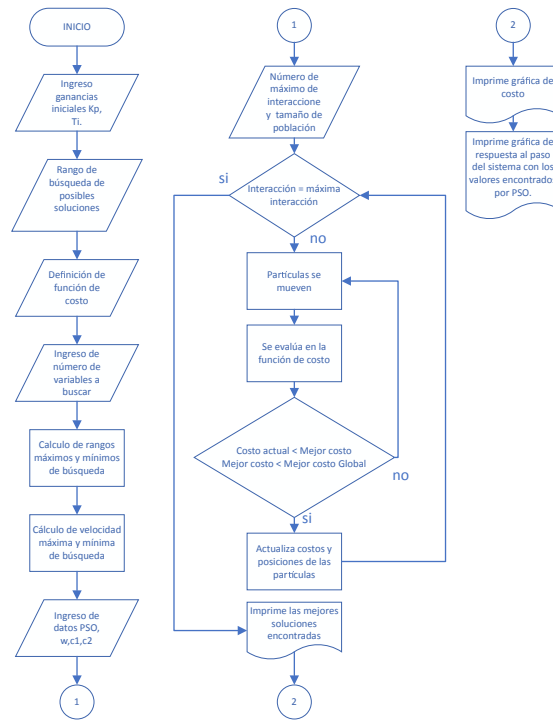


Figure 3.19: Diagrama de flujo de algoritmo PSO implementado.

```

Command Window

Optimizando: 100%  [.....]Completado
-----Valores obtenidos por PSO-----
kp= 12.852
ki= 0.37206
ti= 34.543
preset=0.15949

```

Figure 3.20: Valores obtenidos por el algoritmo PSO en Matlab.

3.2.8 Implementación de controlador PI+CI en tarjeta STM32F407

Comprobado el funcionamiento en las diferentes simulaciones, se procede a la implementación física del controlador, programando la Tarjeta STM32F407 utilizando la librería Waijung desde el simulink de Matlab, como se muestra en la

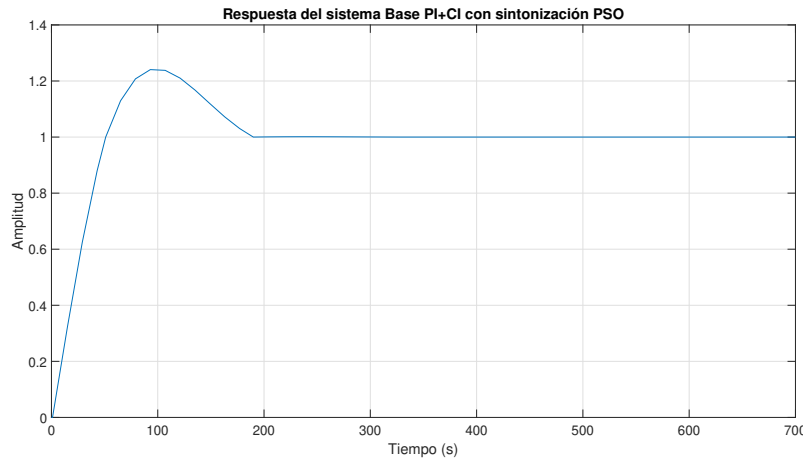


Figure 3.21: Respuesta al paso del sistema EPC con los valores encontrados por el algoritmo PSO

Figura 3.22.

En la programación gráfica de simulink se observa el ingreso del valor de voltaje del sensor de temperatura por una entrada analógica, que es adaptada y convertida a temperatura en grados celcius, tal como en la programación de lazo abierto mostrada en la Figura 3.4, de la misma forma se procede con la lectura del setpoint. Seguidamente, se implementa el modelo de controlador PI+CI tal como en la Figura 3.15, cerrando el lazo de control.

En la Figura 3.22, se observa que el integrador es discreto, esto debido a que la tarjeta STM32F407 no tiene la capacidad de calcular los infinitos valores del modelo continuo; con la misma lógica se establece la frecuencia de trabajo de la STM32F407, que en este caso es de 100Hz, principalmente porque es la frecuencia máxima que soporta el LCD, si se retira el LCD la frecuencia de trabajo puede ser a su máxima capacidad de 8MHz a 25MHz, pero dejando en claro que el PWM debe ser máximo de 100Hz, que es la frecuencia de trabajo máximo que soporta el LED del EPC. Esto último puede configurarse en cada bloque de programación de PWM que se requiera.

Para programar de forma correcta la tarjeta STM32F407, conectada a un puerto usb, se debe pulsar en el botón de conexión como se muestra en la Figura 3.23, bajo ningún concepto de debe pulsar el botón de simulación o el botón

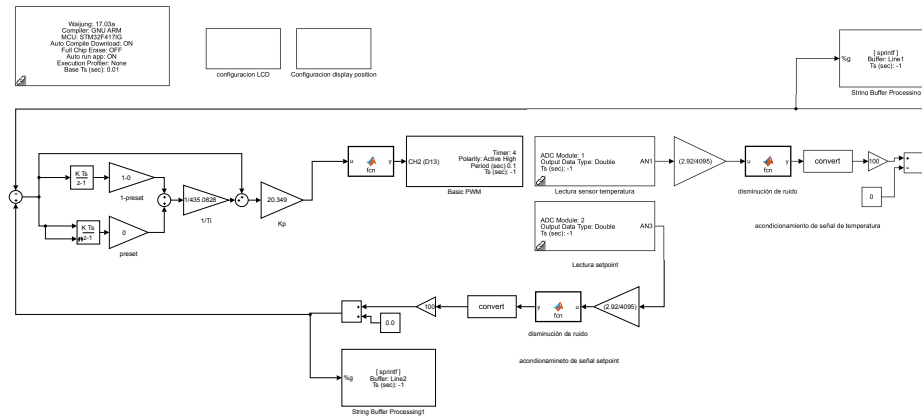


Figure 3.22: Programación de control PI+CI en simulink.

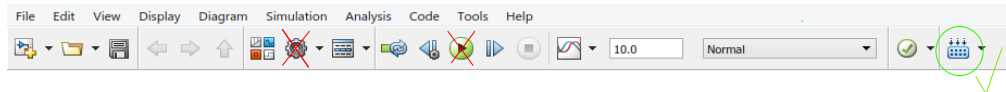


Figure 3.23: Botón de conexión para programar de forma correcta la tarjeta STM32F407.

de configuración (engranaje), porque esto causa un conflicto con el bloque de configuración interno de la programación y arruina el programa. Teniendo que eliminar completamente ese programa y volver a realizar la programación desde cero.

Conectado todo como en la Figura 3.3, se ejecuta el código en LabVIEW mandando un valor de referencia o setpoint, observando como se controla la temperatura de forma gráfica y numérica. Tal como se observa en la Figura 3.24.

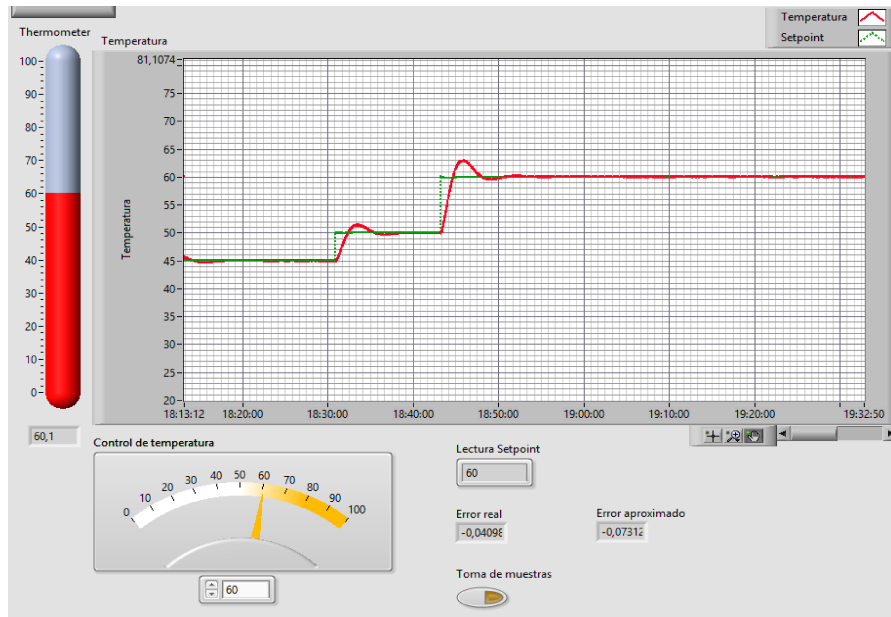


Figure 3.24: Control de temperatura PI+CI implementado, visualización en LabVIEW.

Ahora, ampliando el rango de búsqueda en el algoritmo PSO, se puede encontrar valores que puedan eliminar el sobre impulso, con una buena robustez y tiempo de establecimiento.

3.3 Resultados

Para evaluar el desempeño del control PI+CI sintonizado por PSO, frente al control clásico PI sintonizado usando el aplicativo PIDTuner de matlab, en una implementación práctica, se extraen los datos de cada sintonización realizada, se comparan sus atributos usando el criterio ITAE, y se observa cual de las sintonizaciones muestra un menor valor de ITAE. Se sintoniza el control PI usando la aplicación PIDTuner de Matlab y usando el algoritmo PSO, de manera similar se procede con el control PI+CI, finalmente se amplía el rango de búsqueda del algoritmo PSO con el controlador PI+CI y se muestran todos los resultados obtenidos.

Las ganancias obtenidas por la aplicación PIDTuner de matlab son: $k_p = 10.71$

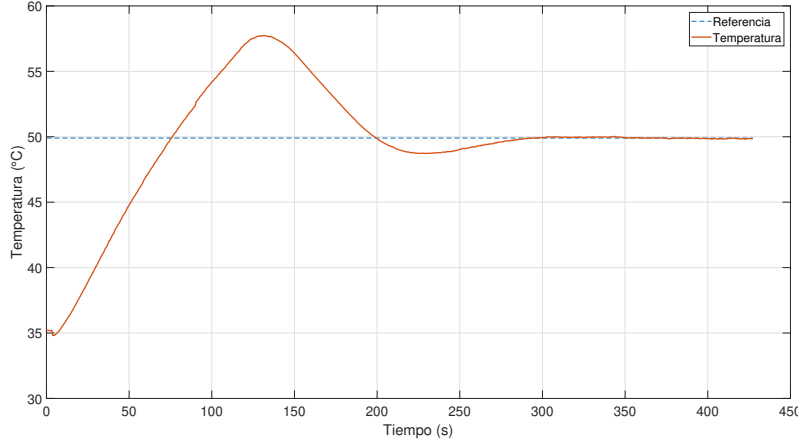


Figure 3.25: Respuesta del sistema implementado con control PI sintonizado con PIDTuner.

y $T_i = 33.12$, la respuesta del sistema en simulación es mostrada en la Figura 3.14, mientras que la respuesta en la implementación física se muestra en la Figura 3.25.

Se observa en la Figura 3.25, que el comportamiento de la implementación del control PI sintonizado usando la aplicación PIDTuner, es muy similar a la simulación realizada. Los valores obtenidos con LabVIEW, y procesados en Matlab, se muestran en color rojo, mientras que la referencia o setpoint en línea segmentada.

En la Figura 3.26, se muestra los datos obtenidos al aplicar el algoritmo PSO para encontrar valores dentro de un rango de búsqueda de $\pm 20\%$ a los valores anteriores $k_p = 10.71$ y $T_i = 33.12$ del PIDTuner. Consiguiendo el algoritmo PSO entregar nuevos valores $k_p = 12.852$ y $T_i = 34.5428$, los cuales se ingresan al controlador en este caso a la tarjeta STM32F407.

Con las ganancias de la aplicación PIDTuner $k_p = 10.71$ y $T_i = 33.12$, se calcula el porcentaje de reseteo, que en esta caso resulta $p_{reset} = 0.18333$. Los resultados de la implementación en el controlador PI se muestran en la Figura 3.27.

El algoritmo PSO con los valores encontrados de $k_p = 12.852$ y $T_i = 34.5428$, calcula automáticamente el p_{reset} del sistema, al hacer uso de su función de evaluación o coste, en este caso el valor obtenido por el algoritmo PSO es $p_{reset} = 0.15544$, que al ingresar al controlador PI, le da la facultad de reseteo del

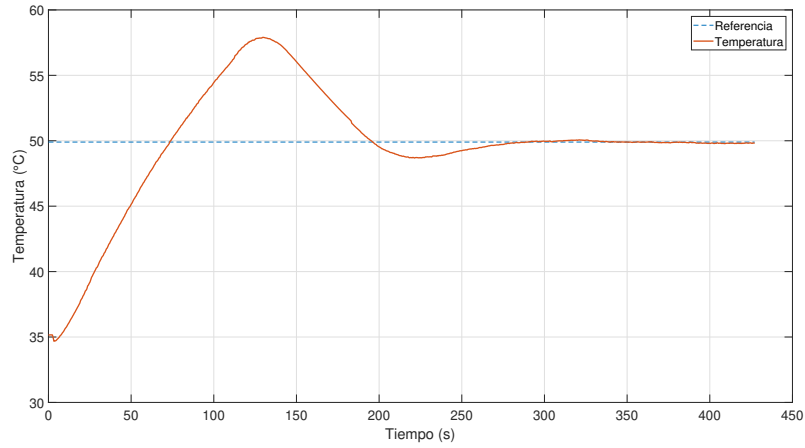


Figure 3.26: Respuesta del sistema implementado con control PI sintonizado con PSO.

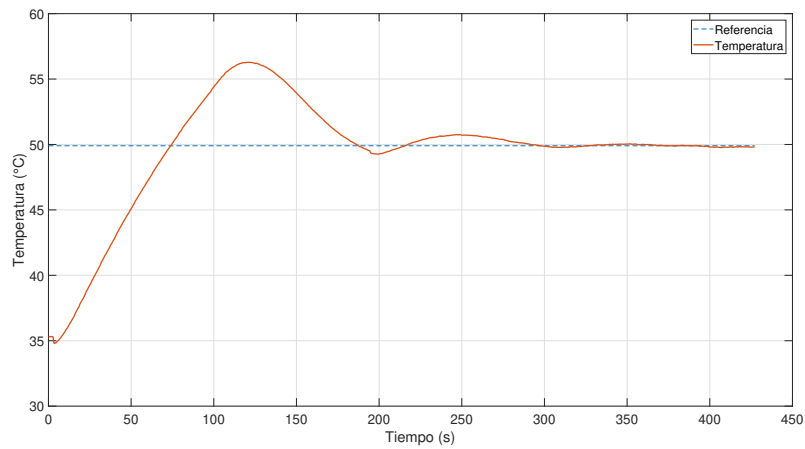


Figure 3.27: Respuesta del sistema implementado con control PI+CI sintonizado con PIDTuner.

integrador de Clegg, específicamente, el porcentaje de reseteo es de 15.544%, los

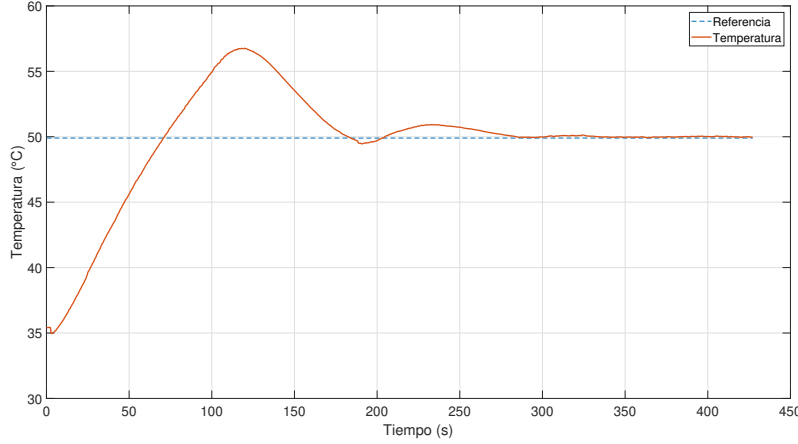


Figure 3.28: Respuesta del sistema implementado con control PI+CI sintonizado con PSO.

resultados de la implementación se observan en la Figura 3.28.

El rango de búsqueda de soluciones aplicado al algoritmo PSO, puede ampliarse o disminuirse, esto se realiza a criterio del desarrollador, teniendo en cuenta que al ampliar el rango de búsqueda de soluciones el algoritmo PSO, tarda mucho mas en encontrar valores adecuados, pero la gran ventaja, es que serán valores que posiblemente puedan lograr un control con muy poco sobre impulso y buen tiempo de establecimiento. Pero también existe el riesgo que los valores que encuentre no necesiten un porcentaje de reseteo o $p_{reset} = 0$, lo cual dejaría la robustez del sistema como un controlador PI tradicional. Con esto último se puede deducir que al ampliar el rango de búsqueda de soluciones, el algoritmo PSO establece si es necesario o no el sistema reseteado de acuerdo a los valores que encuentre y evalúe en la función de costo.

Al ampliar el rango de búsqueda al $\pm 90\%$, de los valores $k_p = 10.71$ y $T_i = 33.12$, se obtienen los siguientes valores de $k_p = 20.349$, $T_i = 435.0828$ y $p_{reset} = 0$. El resultado de la implementación se muestran en la Figura 3.29.

Como análisis final se muestra en la Figura 3.30, un compendio de las respuestas del sistema usando el control PI clásico y el control PI+CI, con los diferentes métodos de sintonía. En la tabla 3.1 se muestra las características de cada implementación.

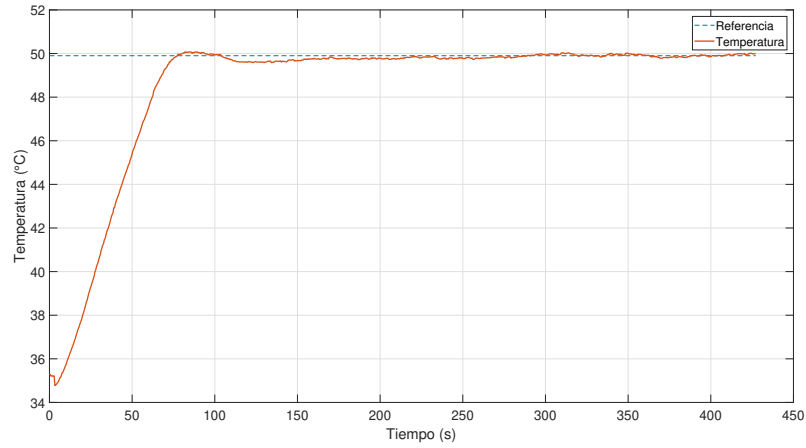


Figure 3.29: Respuesta del sistema implementado con control PI+CI sintonizado con PSO rango ampliado.

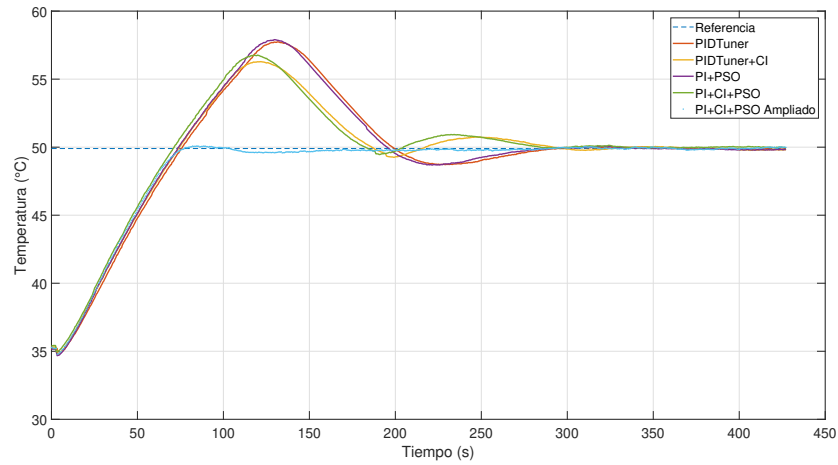


Figure 3.30: Respuesta del sistema implementado con control PI vs control PI+CI.

Table 3.1: Resultados de implementación de diferentes controladores y sintonizaciones

| Tipo de control | Sobre impulso (%) | Tiempo de respuesta (s) | Tiempo de establecimiento (s) | Error en estado estable | ITAE |
|-----------------|-------------------|-------------------------|-------------------------------|-------------------------|-------------|
| PIDTuner | 15.7057 | 52.4318 | 275.1719 | 0.079509 | 104856.4457 |
| PI+PSO | 16.0372 | 51.7025 | 271.0121 | 0.01913 | 108154.2881 |
| PIDTuner+CI | 12.7965 | 51.9333 | 286.2816 | 0.01701 | 85769.7917 |
| PI+CI+PSO | 13.7634 | 49.7111 | 278.548 | 0.034544 | 87228.322 |
| PI+CI+PSO | 0.35624 | 49.5397 | 136.7004 | 0.003154 | 21520.833 |
| Rango Ampliado | | | | | |

3.3.1 Análisis Estadístico.

Con el fin de analizar la eficiencia en el consumo de energía, de un controlador PI+CI frente a un control PI clásico, se realiza un test de Wilcoxon para datos pareados, con el cual se utiliza los valores de ITAE tanto del PI+CI así como del PI clásico, obtenidos de la implementación en el control de temperatura en 31 ocasiones.

Se utiliza el test de Wilcoxon por su facilidad y precisión de 95%, que se puede obtener con número de muestras mayores a 25 [[M^a Purificación Vicente Galindo, 2014](#)].

Los valores de sintonización k_p y T_i obtenidos tanto para el controlador PI clásico así como para el PI+CI cumplen lo descrito para (3.10).

El PIDTuner utiliza el método de Euler el cual se compara con el algoritmo PSO. Para realizar el análisis estadístico con un test de Wilcoxon se siguen los siguientes pasos:

1. Se calculan las diferencias en cada elemento de la muestra pareada.
2. Se eliminan los elementos que tengan diferencias nulas.
3. Se ordena las diferencias prescindiendo del los signos.
4. Asignar un rango de orden a las diferencias.

5. Corregir ligaduras poniendo el mismo rango a valores repetidos.
6. Sumar los rangos según los signos que tengan las diferencias, y obtener los estimadores $T+$ y $T-$.
7. Definir un estadígrafo $T = \min[T+, T-]$
8. Calcular la distribución normal usando (3.11)
9. Comparar el valor obtenido con una constante de confianza $Z_{\alpha}=1.96$.
10. Se toma la decisión final en base a los valores obtenidos.

$$Z = \frac{T - n \frac{(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}} \quad (3.11)$$

La Ecuación (3.11) se usa para calcular la distribución normal del estadígrafo.

En la Tabla 3.2, se muestra la hipótesis nula o principal y la hipótesis alternativa, así como los valores ITAE obtenidos al implementar el control PI clásico sintonizado por PIDTuner de Matlab, y los valores ITAE del control PI+CI sintonizado por algoritmo PSO. Seguidamente en la Tabla 3.3, se muestra el análisis de los datos, el cálculo estadístico y la decisión tomada en base a los resultados obtenidos.

3.4 Discusión.

Al analizar los resultados obtenidos, tanto en la implementación física del controlador como el análisis estadístico, se observan las ventajas del controlador PI+CI+PSO frente a la aplicación interna PIDTuner de Matlab de sintonización de controladores clásicos, pero aún debe hacerse una comparación general con otras implementaciones.

Como se menciona en el Capítulo 2, en el Estado del Arte, diferentes autores realizan control de temperatura usando diferentes métodos, cada uno obteniendo características muy llamativas, como: setpoint utilizado, Tiempo de subida, Tiempo de establecimiento, Porcentaje de sobre impulso, Valor de sobre impulso y error en estado estable. Si se compara esas características con los obtenidos con el control PI+CI desarrollado en este estudio, se tiene la Tabla 3.4.

Como se puede observar en la Tabla 3.4, es un resumen de las diferentes implementaciones de controles de temperatura analizadas brevemente en el Capítulo 2. Los controles 1, 2 y 3 de la Tabla 3.4, propuestos en [Tsai et al., 2017], muestran los controladores predictivos.

- El control PID+EP que tiene buen tiempo de subida (10s), pero alto sobre impulso (30.89%), con un tiempo de estabilización grande de alrededor de 95s.

- El control PID+RGA que tiene mayor tiempo de subida que el anterior (12.65s), pero tiene menor sobre impulso (11.9%) y menor tiempo de establecimiento (91s).
- El control PI+PSO+RGA que tiene menor tiempo de subida (12.51s), menor sobre impulso que las dos anteriores (8.32%) y tiempo de estabilización de 79s.

Los controles 4 y 5 de la Tabla 3.4, propuestos en [Wang et al., 2017a], ofrecen los siguientes resultados:

- Control PID clásico que tiene un tiempo de subida de 60s, un tiempo de establecimiento de 400s un porcentaje de sobre impulso de 30%.
- Control PID de neurona simple, con un tiempo de subida de 370s, un tiempo de establecimiento de 700s y un porcentaje de sobre impulso del 3%.

En la implementación realizada en [Hernandez-Ambato et al., 2018], muestra tres tipos de controladores 6, 7 y 8 de la Tabla 3.4:

- Control on/off con un tiempo de subida de 850s, un tiempo de establecimiento de 1400s, un porcentaje de sobre impulso de 4.29% y un error en régimen permanente de 4.62°C.
- Control PI clásico con un tiempo de subida de 2400s, un tiempo de establecimiento de 6400s, un sobre impulso de 2.86% y un error de régimen permanente de 0.2985°C.
- Control PI anti-windup con un tiempo de subida de 1000s, un tiempo de establecimiento de 1600s, un porcentaje de sobre impulso de 3.57% y un error de régimen permanente de 0.12°C.

Los controles 9 y 10 de la Tabla 3.4, implementados en citepqiao2018application, muestran el uso de control fuzzy. El parámetro de diseño más importante que indican los autores es, el error en régimen permanente que no puede ser mayor a $\pm 0.1^\circ\text{C}$ para un punto de referencia de 45°C obteniéndose finalmente una precisión muy elevada de $\pm 0.01^\circ\text{C}$.

En el control 11 de la Tabla 3.4, implementado en [Salamea et al., 2018], se observa que trabajan en niveles de voltaje de 0V a 5V con una relación de temperatura $0^\circ\text{C} > 1\text{V}$ y $330^\circ\text{C} > 5\text{V}$ [Salamea et al., 2018]. Obteniendo un tiempo de subida muy rápido 0.6s, un tiempo de establecimiento de 4s pero con un sobre impulso del 30%, obteniendo un error en estado estable de $\pm 5^\circ\text{C}$.

Los controles implementados en [Dulău et al., 2018], se incluyen en los numerales 12 y 13 de la Tabla 3.4, mostrando que a pesar de tener bajo sobre impulso del 3%, su error en estado estable es al rededor de los $\pm 4^\circ\text{C}$.

Los controladores implementados en [Oo et al., 2019], son mostrados en los numerales 14, 15 y 16 de la Tabla 3.4, muestra un control sin sobre impulso y con un tiempo de establecimiento de 8.4s.

Los últimos 3 controladores de la Tabla 3.4, corresponden a los numerales 17, 18 y 19, pertenecen a los implementados en este estudio, indicando que para un setpoint de $49.9^{\circ}C$:

- El controlador sintonizado con PIDTuner de Matlab, se tiene 73.6s en el tiempo de subida, 386.1s en el tiempo de establecimiento, con un porcentaje de sobre impulso de 15.7% y un error en estado estable de $\pm 0.08^{\circ}C$.
- El controlador implementado PI+CI+PSO sintonizado con un porcentaje de búsqueda de $\pm 20\%$ para cada ganancia, se tiene 60s en el tiempo de subida, 336.5s en el tiempo de establecimiento, con un porcentaje de sobre impulso de 13.8% y un error en estado estable de $\pm 0.05^{\circ}C$.
- El controlador implementado PI+CI+PSO sintonizado con un porcentaje de búsqueda de $\pm 90\%$ para cada ganancia, se tiene 66s en el tiempo de subida, 183.34s en el tiempo de establecimiento, con un porcentaje de sobre impulso de 0.36% y un error en estado estable de $\pm 0.1^{\circ}C$.

3.5 Conclusiones y recomendaciones.

Como se observa en las diferentes implementaciones realizadas, se puede decir que el controlador PI+CI es superior al control tradicional PI, en margen de fase, es decir mejora la robustez del sistema sin sacrificar tiempo de establecimiento, como se observa en la Figura 3.30, pero esto a su vez, en sistemas que tengan sobre oscilaciones o sean muy oscilantes, que por muchos métodos de sintonía que se apliquen no se puede mejorar mas. En ese caso, es recomendable utilizar un controlador PI+CI, en lo que se refiere a sintonía es recomendable usar algoritmos como el PSO, que facilitan mucho la tarea de sintonización, optimización y ahorran tiempo.

Aunque teóricamente se indica que el reseteo completo se da luego del segundo cruce por cero del error, en la práctica se observa que el reseteo es notorio a partir del tercer cruce por cero, esto debido al ruido y a la baja estabilidad del setpoint.

El setpoint no es un número fijo, debido a las variaciones de voltaje que siempre existen en los sistemas electrónicos, por lo cual tiene su propia función de transferencia, y genera una respuesta del sistema más oscilante de lo previsto.

Para controladores con poco sobre impulso y muy poco oscilantes, no será mayor el cambio si se utiliza sistemas reseteados.

Al ampliar el rango de búsqueda de valores para el algoritmo PSO en la planta EPC, el algoritmo PSO puede determinar que no es necesario el porcentaje de reseteo para una mejor optimización, por lo cual el controlador de clegg es eliminado, dejando solo un control clásico PI.

Debido al análisis estadístico se demuestra que el controlador PI+CI sintonizado por PSO es mejor que el control clásico PI sintonizado por PIDTuner de matlab, en el proceso de control de temperatura, pues logra disminuir las pérdidas de energía en el sistema.

Al realizar la comparación del control PI+CI+PSO frente a otras implementaciones, se observa que los resultados son bastante aceptables, por lo que es un control muy robusto y con muy buenas prestaciones para ser tomado en cuenta en diferentes implementaciones de control de temperatura.

Table 3.2: Muestras de valores ITAE obtenidas para análisis de test de Wilcoxon en el proceso de control de temperatura usando PI+CI

| TEST DE WILCOXON PARA DATOS PAREADOS, EN EL PROCESO DE CONTROL DE TEMPERATURA | | | | | | | | | |
|---|---|--------------|------------|----------|----|----|----------|--------------|-------|
| Hipótesis | Principal(Ho): $ITAE_{EULER} = ITAE_{PSO}$ $Media(ITAE_{Euler}) = Media(ITAE_{PSO})$ | | | | | | | | |
| | Alternativa(Ha): $ITAE_{EULER} > ITAE_{PSO}$ $Media(ITAE_{Euler}) > Media(ITAE_{PSO})$ | | | | | | | | |
| n | $ITAE_{EULER}$ | $ITAE_{PSO}$ | DIFERENCIA | ASIGNADO | T+ | T- | LIGADURA | DIF ORDENADA | RANGO |
| 1 | 63325,035 | 33191,856 | 30133,179 | 8 | 8 | - | | 7480,08 | 1 |
| 2 | 171092,368 | 29689,435 | 141402,932 | 19 | 19 | - | | 9325,44 | 2 |
| 3 | 336606,952 | 80380,929 | 256226,023 | 26 | 26 | - | | 11667,13 | 3 |
| 4 | 250753,638 | 33235,119 | 217518,519 | 23 | 23 | - | | 11708,04 | 4 |
| 5 | 124823,751 | 20341,331 | 104482,420 | 17 | 17 | - | | 17781,71 | 5 |
| 6 | 355523,182 | 46728,062 | 308795,119 | 30 | 30 | - | | 18593,46 | 6 |
| 7 | 329066,964 | 44307,161 | 284759,803 | 29 | 29 | - | | 26969,11 | 7 |
| 8 | 299161,891 | 61435,022 | 237726,869 | 24 | 24 | - | | 30133,18 | 8 |
| 9 | 289225,234 | 25150,128 | 264075,107 | 28 | 28 | - | | 32148,91 | 9 |
| 10 | 351357,052 | 29810,483 | 321546,568 | 31 | 31 | - | | 35164,20 | 10 |
| 11 | 295838,907 | 38652,740 | 257186,167 | 27 | 27 | - | | 41308,92 | 11 |
| 12 | 300603,558 | 44499,074 | 256104,484 | 26 | 26 | - | | 43957,28 | 12 |
| 13 | 204808,055 | 43889,677 | 160918,377 | 20 | 20 | - | | 47501,35 | 13 |
| 14 | 164154,487 | 45678,135 | 118476,352 | 18 | 18 | - | | 59778,75 | 14 |
| 15 | 126690,063 | 26661,662 | 100028,401 | 16 | 16 | - | | 60032,72 | 15 |
| 16 | 89857,406 | 29824,683 | 60032,723 | 15 | 15 | - | | 100028,40 | 16 |
| 17 | 88292,099 | 28513,347 | 59778,752 | 14 | 14 | - | | 104482,42 | 17 |
| 18 | 75788,585 | 28287,232 | 47501,353 | 13 | 13 | - | | 118476,35 | 18 |
| 19 | 65945,352 | 21988,070 | 43957,281 | 12 | 12 | - | | 141402,93 | 19 |
| 20 | 63248,374 | 21939,457 | 41308,918 | 11 | 11 | - | | 160918,38 | 20 |
| 21 | 58085,574 | 25936,661 | 32148,913 | 9 | 9 | - | | 164304,41 | 21 |
| 22 | 52338,336 | 17174,138 | 35164,198 | 10 | 10 | - | | 167134,59 | 22 |
| 23 | 47687,634 | 20718,524 | 26969,111 | 7 | 7 | - | | 217518,52 | 23 |
| 24 | 39319,856 | 20726,393 | 18593,463 | 6 | 6 | - | | 237726,87 | 24 |
| 25 | 38172,848 | 20391,137 | 17781,711 | 5 | 5 | - | 26 | 256104,48 | 25 |
| 26 | 31448,873 | 19740,834 | 11708,039 | 4 | 4 | - | 26 | 256226,02 | 26 |
| 27 | 29015,209 | 19689,772 | 9325,437 | 2 | 2 | - | | 257186,17 | 27 |
| 28 | 28803,734 | 17136,602 | 11667,132 | 3 | 3 | - | | 264075,11 | 28 |
| 29 | 27329,679 | 19849,598 | 7480,082 | 1 | 1 | - | | 284759,80 | 29 |
| 30 | 177103,397 | 12798,990 | 164304,407 | 21 | 21 | - | | 308795,12 | 30 |
| 31 | 177103,397 | 9968,808 | 167134,588 | 22 | 22 | - | | 321546,57 | 31 |

Table 3.3: Análisis de resultados usando test de Wilcoxon.

| | |
|--|--|
| Total Suma T+ | 497 |
| Total Suma T- | - |
| $T = \min(T+, T-)$ | 497 |
| $Z = \frac{T - n \frac{(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}}$ | 4,879539715 |
| Nivel de confianza del 95% (Z_a) | 1.96 |
| Decisión | |
| $Z \leq Z_a$ se acepta H_0 | - |
| $Z > Z_a$ se acepta H_a | Ok |
| Interpretación | Como los valores ITAE del método de euler del PIDTuner son mayores al ITAE del algoritmo PSO, se concluye que el controlador PI+CI sintonizado por PSO es mejor que el control clásico PI sintonizado por PIDTuner de matlab, en el proceso de control de temperatura, pues logra disminuir las pérdidas de energía en el sistema. |

Table 3.4: Resumen de implementación de diferentes controles de temperatura.

| Controles de temperatura implementados por diferentes autores. | | | | | | | | |
|--|-----------------|----------------------|----------------------|-------------------------------|------------------------------|------------------------------------|---|--|
| N° | Tipo de Control | Setpoint $^{\circ}C$ | Tiempo de subida (s) | Tiempo de establecimiento (s) | Porcentaje Sobre impulso (%) | Pico sobre impulso ($^{\circ}C$) | Máximo error permitido ($\pm^{\circ}C$) | Error en estado estable ($\pm^{\circ}C$) |
| 1 | PID+EP | 175 | 10 | 95 | 30,89 | 21,87 | 3 | - |
| 2 | PID+RGA | 175 | 12,65 | 91 | 11,9 | 3,125 | 3 | 2,22 |
| 3 | PI+PSO+RGA | 175 | 12,51 | 79 | 8,32 | 2 | 3 | 1,37 |
| 4 | PIDclasico | 100 | 60 | 400 | 30 | 30 | 3 | - |
| 5 | PIDneurona | 100 | 370 | 700 | 3 | 3 | 3 | - |
| 6 | ON/OFF | 140 | 850 | 1400 | 4,29 | 6 | 3 | 4,62 |
| 7 | PIclasico | 140 | 2400 | 6400 | 2,86 | 4 | 3 | 0,2985 |
| 8 | PI+antiwindup | 140 | 1000 | 1600 | 3,57 | 5 | 3 | 0,12 |
| 9 | PIDclasico | 45 | 2400 | 12000 | 5 | 0,5 | 1 | 0,5 |
| 10 | Fuzzy+PID | 45 | - | - | - | - | 1 | 0,01 |
| 11 | PID | 70 | 0,6 | 4,1 | 30 | 29 | - | 5 |
| 12 | ON/FF | 25 | 81,6 | 86,4 | 3,3 | 0,83 | 4 | - |
| 13 | ON/FF | 35 | 306,6 | 322,8 | 3 | 0,83 | 4 | - |
| 14 | PIDclasico | 1 | 2,7 | 8 | 10 | 0,1 | - | - |
| 15 | FuzzyPID | 1 | 3,9 | 8 | 3 | 0,03 | - | - |
| 16 | Fuzzy | 1 | 8,4 | 8,4 | 0 | 0 | - | - |
| 17 | PIDTuner | 49,9 | 73,6 | 386,1 | 15,7 | 7.8 | - | 0,08 |
| 18 | PI+CI+PSO | 49,9 | 60 | 336,5 | 13,8 | 6.85 | - | 0,05 |
| 19 | PI+CI+PSO | 49,9 | 66 | 183,34 | 0,36 | 0.18 | - | 0,1 |

Bibliography

- L. Aimagin Co. Waijung Blockset, 2017. URL <https://www.aimagin.com/>.
- K. J. Åström and R. M. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.
- A. Baños Torrico, Á. y. Vidal Sánchez, and J. Carrasco Gómez. Compensador para un sistema de control realimentado y procedimiento para sintonizar dicho compensador., 2010.
- A. Barreiro and A. Baños. Sistemas de control basados en reset. *Revista Iberoamericana de Automática e Informática industrial*, 9(4):329–346, 2012.
- C. P. Basso. *Linear circuit transfer functions: An introduction to fast analytical techniques*. John Wiley & Sons, 2016.
- O. Beker, C. V. Hollot, Y. Chait, and H. Han. Fundamental properties of reset control systems. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 15(1):187–192, 2002. ISSN 14746670. doi: 10.3182/20020721-6-ES-1901.00112.
- O. Beker, C. Hollot, Y. Chait, and H. Han. Fundamental properties of reset control systems. *Automatica*, 40(6):905–915, 2004.
- J. Campos, S. Jaramillo, L. Morales, O. Camacho, D. Chávez, and D. Pozo. Pso tuning for fuzzy pd+ i controller applied to a mobile robot trajectory control. In *2018 International Conference on Information Systems and Computer Science (INCISCOS)*, pages 62–68. IEEE, 2018.
- J. Clegg. A nonlinear integrator for servomechanisms. *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, 77(1):41–42, 1958.
- A. G. Daful. Comparative study of pid tuning methods for processes with large & small delay times. In *2018 Advances in Science and Engineering Technology International Conferences (ASET)*, pages 1–7. IEEE, 2018.

- J. Doktian, W. Pongyart, and P. Vanichchanunt. Development of a semi auto-tuning algorithm for pi+ ci reset controller. In *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, pages 155–158. IEEE, 2019.
- C. Duarte and J. Quiroga. System identification of a dc motor using pso algorithm. *Revista Facultad de Ingeniería Universidad de Antioquia*, 1(55):116–124, 2010.
- M. Dulău, M. Karoly, and T. M. Dulău. Fluid temperature control using heat exchanger. *Procedia Manufacturing*, 22:498–505, 2018. ISSN 23519789. doi: 10.1016/j.promfg.2018.03.058. URL <https://doi.org/10.1016/j.promfg.2018.03.058>.
- M. A. El-Shafei, M. I. El-Hawwary, and H. M. Emara. Implementation of fractional-order pid controller in an industrial distributed control system. In *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*, pages 713–718. IEEE, 2017.
- J. I. Fernández. *Controladores PI con acción de reset*. PhD thesis, Universitat Autònoma de Barcelona, 2011.
- J. Hernandez-Ambato, J. Rodriguez-Flores, J. Cortes-Llanganate, and F. Cabrera-Aguayo. Classic controllers design applied to temperature control for a plastic thermoforming machine. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 551–558. IEEE, 2018.
- I. Horowitz and P. Rosenbaum. Non-linear design for cost of feedback reduction in systems with large parameter uncertainty. *International Journal of Control*, 21(6):977–1001, 1975.
- S. Joraked, W. Pongyart, and K. Angkeaw. Design and implementation of a fpaa based pi+ ci reset controller emulator. In *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, pages 266–269. IEEE, 2018.
- W. M. Montalvo López. Diseño y construcción de un controlador pid auto-sintonizable mediante inteligencia de enjambre para determinar el comportamiento colectivo de los sistemas pertenecientes a procesos estudiados en el cerfín-secap. Master’s thesis, Quito, 2015., 2015.
- M^a Purificación Vicente Galindo. Estadística para investigadores: todo lo que quiso saber y nunca se atrevió a preguntar, 2014. URL <https://miriadax.net/web/estadistica-investigadores-2edicion>.
- National Instruments. ENTRENADOR DE PLANTA DE CONTROL “EPC”. Technical report, National Instruments Ecuador, 2017. URL www.datalights.com.ec.

- H. L. Oo, S. Anatolii, and K. Z. Ye. Analysis and evaluation of the efficiency of laser temperature control system. In *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 2180–2182. IEEE, 2019.
- O. A. Orozco and V. M. A. Ruiz. Sintonización de controladores pi y pid utilizando los criterios integrales iae e itae. *Revista Ingeniería*, 13(1-2):31–39, 2003.
- C. Prieur, I. Queinnec, S. Tarbouriech, and L. Zaccarian. Analysis and synthesis of reset control systems. *Foundations and Trends in Systems and Control*, 6(2-3): 117–338, 2018.
- Y. Qiao, J. Pengt, W. Ye, Q. Li, Y. Yang, X. Sun, and D. Tang. Application of improved intelligent pid algorithm in high precision thermostatic control in trace water analyzer. In *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 440–447. IEEE, 2018.
- V. M. A. Ruíz. Métodos de sintonización de controladores pid que operan como reguladores. *Revista Ingeniería*, 12(1-2):21–36, 2002.
- H. T. Salamea, N. C. Guerrero, S. R. Ávila, and O. Alvarado-Cando. Analysis and modeling of temperature control for the crude mill of a cement plant: An industrial case study involving pid control. In *2018 IEEE ANDESCON*, pages 1–6. IEEE, 2018.
- N. Selamat, N. Wahab, and S. Sahlan. Particle swarm optimization for multivariable pid controller tuning. In *2013 IEEE 9th International Colloquium on Signal Processing and its Applications*, pages 170–175. IEEE, 2013.
- Y. Tian, Q. Wang, Y. Wang, and Q. Jin. A novel design method of multi-objective robust pid controller for industrial process. In *2014 9th IEEE Conference on Industrial Electronics and Applications*, pages 242–246. IEEE, 2014.
- C.-C. Tsai, F.-C. Tai, and R.-S. Liu. Intelligent predictive temperature control using pso-rga for transfer mold heating processes in semiconductor die packaging machines. In *2017 International Conference on System Science and Engineering (ICSSE)*, pages 510–514. IEEE, 2017.
- Z. Wang, X. Li, and X. Lu. Temperature control based on a single neuron pid algorithm for a blackbody radiation source. In *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 220–225. IEEE, 2017a.
- Z. Wang, X. Li, and X. Lu. Temperature control based on a single neuron pid algorithm for a blackbody radiation source. In *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 220–225. IEEE, 2017b.